Video Coding
for
Compression and Content-Based Functionality

Thesis submitted for the degree of

Doctor of Philosophy

by

Patrick Joseph Mulroy BEng AMIEE

Dublin City University
Supervisor : Dr. Seán Marlow
School of Electronic Engineering

May 1999

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Pat McIlroy ID No.: 93701403

Date: 29th May 1999

Dedicated to my parents ...

Acknowledgements

I am very grateful to a number of people for their encouragement and support throughout this undertaking.

First and foremost my supervisor, Dr. Seán Marlow, for his help with the direction of this research work, his continued encouragement and his patience over nearly five years. Also, the rest of the video coding group at Dublin City University, especially the three Noels - Dr. Noel O'Connor, Dr. Noel Brady, Dr. Noel Murphy and Liam Ward who along with the other colleagues at MPEG and COST made it all more fun!

I also worked with a number of other MPEG, ITU-T and COST colleagues, on the development of video coding standards simulation and segmentation software. I would like to particularly thank Karl Olav Lillevold of Telenor Research for his assistance on the integration of the arithmetic coding work into the H.263 simulation software. Also Paulo Villegas of Telefonica I+D for assistance with the RSST work within the COST 211^{quat} Analysis Model. A number of BT colleagues also helped at various stages of this work, in particular, Wayne Ellis, Mark Shackleton, Mike Nilsson, Dr. Mark Beaumont and Dr. Bill Welsh.

I am very much indebted to my former group leader of the Image Coding and Processing Group, Mike Whybray, and my employer, BT Labs, for sponsoring this part time Ph.D. Mike's encouragement and support at the right times was very appreciated.

Finally, thanks also to my family and friends who wondered when I would ever finish!

Contents

Abstract

The lifetime of this research project has seen two dramatic developments in the area of digital video coding. The first has been the progress of compression research leading to a factor of two improvement over existing standards, much wider deployment possibilities and the development of the new international ITU-T Recommendation H.263. The second has been a radical change in the approach to video content production with the introduction of the content-based coding concept and the addition of scene composition information to the encoded bit-stream. Content-based coding is central to the latest international standards efforts from the ISO/IEC MPEG working group.

This thesis reports on extensions to existing compression techniques exploiting *a priori* knowledge about scene content. Existing, standardised, block-based compression coding techniques were extended with work on arithmetic entropy coding and intra-block prediction. These both form part of the H.263 and MPEG-4 specifications respectively. Object-based coding techniques were developed within a collaborative simulation model, known as SIMOC, then extended with ideas on grid motion vector modelling and vector accuracy confidence estimation. An improved confidence measure for encouraging motion smoothness is proposed.

Object-based coding ideas, with those from other model and layer-based coding approaches, influenced the development of content-based coding within MPEG-4. This standard made considerable progress in this newly adopted content based video coding field defining normative techniques for arbitrary shape and texture coding. The means to generate this information, the analysis problem, for the content to be coded was intentionally not specified. Further research work in this area concentrated on video segmentation and analysis techniques to exploit the benefits of content based coding for generic frame based video. The work reported here introduces the use of a clustering algorithm on raw data features for providing initial segmentation of video data and subsequent tracking of those image regions through video sequences. Collaborative video analysis frameworks from COST 211^{quat} and MPEG-4, combining results from many other segmentation schemes, are also introduced.

1. Introduction

1.1 Digital Video Coding

Digital video has become increasingly important with the recent proliferation of the internet, digital television and general broad and mid-band digital telecommunications systems. Storage and broadcast video applications, now in the digital arena, are already benefiting greatly from video coding technology and this year hundreds of new digital television channels are to be launched throughout the world. Video conferencing and telephone applications, requiring switched and multi-point long distance transmission, and still suffering from a widespread public acceptance issue, were only made possible at all with digitised audio-visual data streams and standardised coding algorithms. Internet browsers, meanwhile, with software decoders, now integrate streaming video to the desktop with picture quality scaling to the available bandwidth. Digital video benefits are recognised but it is only relatively recently, with advances in video coding, that we have learnt how to capitalise on this technology and make many of these applications realistic and economically attractive to use.

Broadcast video, in the digital domain, is encoded according to the sampling rates and quantisation law specified in ITU-R Recommendation BT.601 [1] (formerly known as CCIR-601) and requires a raw digital bit rate of greater than 200 Mbit/s. To put this in context, a standard 1.44 Mbyte computer diskette would hold a small fraction of a second of broadcast quality digital video and even a 650 Mbyte CD-ROM would hold less than half a minute of video in uncompressed form. Household video on demand would not be possible with less than three thousand ISDN digital phone lines installed. These bandwidth restrictions and the processing power required to manipulate the sheer quantities of data involved are the main reasons why the take up of digital video has been so slow.

Computer speeds and general processing power have steadily increased over recent years and this trend shows little sign of stopping at least in the near future. Advances have also been made in both audio and video compression algorithm development and internationally agreed standards are now in place for storage, broadcast and real-time transmission applications. Typically these allow storage of whole movies on CD-ROM, coding at 1.5 Mbit/s, and multipoint communication over ISDN (64 kbit/s to 1920 kbit/s), PSTN (28.8 kbit/s) and even mobile GSM (9.6 kbit/s) channels. Filtering and sub-sampling of the source video is carried out for video at these rates, both spatially, typically to one quarter of BT.601 for so called common intermediate format CIF, or even quarter of that again to QCIF, and also temporally, to between 5 and 15 frames per second. This still represents an impressive compression ratio of between 200 to 500 to 1 but it is important to realise that this level of compression is not achieved losslessly. A marked reduction in fidelity of the image, particularly at the lower rates, is another factor in the public acceptance and adoption of digital video in the videoconferencing and home videophone arenas.

1.2 Background and Motivation for Research

1.2.1 Further Compression

A key motivation for research was improved compression. The standard video compression algorithms in use today are based on a number of statistical and perceptual redundancy reduction techniques. Some statistical techniques such as entropy coding and the Ziv-Lempel algorithm [2] are borrowed from general data compression theory and are in widespread use in file compression utilities and data modems. Others exploit the particular properties of video pictures (e.g. adjacent pixels having similar values and subsequent frames being very similar). All the above techniques can be termed lossless, i.e. the input source video waveform can be reproduced exactly, but compression ratios obtainable are not very high, typically less than 5:1. Perceptual redundancy reduction is much more effective for our purpose but entails lossy coding where only an approximation of the input signal is reproduced after coding. The goal is to ensure that any degradations are in less psychovisually important areas of the signal. Sub-sampling, colour spatial resolution reduction, non-linear quantisation of prediction errors and transform coding are the main techniques in use here. Sub-sampling reduces the actual number of input samples used for coding. Missing samples in the resulting decompressed video can be interpolated from those coded. Colour sensitivity is related to the ratios of rods and cones in our own eyes. We have evolved higher visual acuity for luminance information than for colour so the spatial resolution of the colour information in the signal can be further reduced without a perceived loss of quality. Further quantisation, particularly of predictive error images, can also be subjectively acceptable due to a psycho-visual phenomenon known as *spatial masking* where the eye is less sensitive to errors in high details regions. Transform coding is the final and most effective technique in wide use. Here blocks of image data are mathematically transformed, using the *Discrete Cosine Transform* or DCT, from highly correlated spatial data to highly uncorrelated frequency domain data. This transformation does not in itself provide compression but the transform coefficients are now much more amenable to compression. The coefficients can be reordered in terms of frequency, so called *zig-zag coding*, before further quantisation and entropy coding. Typically the highest spatial frequency components reduce to zero and need not be coded.

All the above techniques have been heavily researched and improved over perhaps three decades now and have been embodied in all the current video coding standards, namely ITU-T H.261, H.263, ISO/IEC MPEG-1 and MPEG-2. However, compression rates for a given sequence fidelity have reached an asymptote where purely statistical methods of video coding are employed. Any further improvement in compression requires a deeper consideration of the nature of video and an abstraction of the problem. All the algorithms and techniques, quoted thus far, regard video as a three-dimensional signal with spatial x and y and temporal t co-ordinates. Arrays of pixel elements are processed regardless of the content within that array, although the consequences of the content are apparent in the resulting coded bitstream. No higher level appreciation of the content is made. The analogy drawn is with the facsimile machine. Text written on a page is coded a raster line at a time by a fax machine. The text itself could be represented as ASCII code and compressed and transmitted much more efficiently but this would require the machine to recognise what it was sending. Video coding at the outset of this research was at the level of the fax machine.

Harashima et al. [3], amongst others, defined a hierarchy of video coding phases ranging from the lowest level of straight pulse code modulation (PCM) coding through what he calls second generation coding schemes such as “Structure/Feature Extraction Coding”. And onwards to third generation “Analysis/Synthesis Coding” and “Intelligent Coding” schemes where decoders would employ inference as necessary. One specialised example of third generation analysis/synthesis coding for videophone application is *Model Based Coding* described by Welsh [4]. Here the underlying model is of a 3D head and shoulders wire-frame model, an extension of the *Candide* [5] head wire-frame described by Forcheimer [6], with support for natural movement, facial expressions etc. The synthesis part of such a coder can give surprisingly realistic results, particularly when real facial texture is mapped on to the wire-frame. The analysis part, tracking of a real person's facial expressions, motion and mouth shape accurately enough to drive the model, remains only partly solved.

A second example of analysis/synthesis coding is the *Layered Coding* approach proposed by the MIT Media Lab and described by Wang et al. [7]. This approach uses mid-level vision techniques, global motion estimation and segmentation to derive a representation of the video content in terms of overlapping layers. Each layer represents content data at different depths in the scene and once obtained can be re-composed into the original scene by direct layered composition. The analysis problem in the form of sequence layer decomposition, is again the hardest part of this process but is shown to work well on a real video sequence albeit one exhibiting well defined coherent motion layers.

The origin of much of the work in this thesis lies in early efforts by a number of researchers studying yet another but much more general third generation analysis/synthesis technique known as *Object-Based Coding* and described by Musmann et al. [8]. This was a new form of so-called "knowledge based" video coding which aimed to segment a picture frame into a number of objects of arbitrary shape and predict parameters of these objects (motion, texture, shape) in subsequent frames. Better prediction of this parameter set would mean much higher coding efficiency, but this relied to a large degree on the chosen coding objects mapping closely to real world objects in the scene. Musmann actually describes two approaches; one based on a source model of moving flexible 2D objects, a superset of traditional block coders, and another based on moving 3D objects. The real objects, of course, live in a three dimensional world and we are coding regions, or projections of the 3D world onto 2D frames. The regions can, however, be considered to fit either underlying model.

1.2.2 Video Content Definition

The motivation thus far has centred on improved compression through the abstraction of video away from blocks of data towards collections of regions or objects. If coded regions relate to semantically meaningful objects in the scene, we should be able to find more compact parameter sets of motion and texture for these regions and improve coding efficiency. But we are also now in a position for the first time to describe the content within a video sequence for other purposes. One application is content manipulation in the coded domain i.e. selection of a character in a scene and placement into an alternative scene. Other examples include preferential treatment of bits associated with this important character when viewing the video over a very limited bandwidth channel, or, association of other information with an object, such as subtitling, that follows the character about the screen. These applications are the focus of the emerging ISO/IEC MPEG-4 [9] coding standard.

Yet another particularly useful application would relate to representation of video content for *searching* and *browsing* purposes. This is a very real need with the rapid deployment of digital multimedia databases including sound and images on the World Wide Web, video on demand systems, and corporate image databases. ISO/IEC are also addressing this area with the upcoming MPEG-7 - *Multimedia Content Description Interface* activity and experimentation model (XM) development [10]. Here, a set of multimedia content descriptors or features, essentially content meta-data, will be specified and used to annotate existing bit-streams.

Whilst much work has now been done in a number of areas required for content based coding of video, mostly by the MPEG-4 group, including shape coding, arbitrary shape DCT, layer composition etc., much work remains to be done in exploiting these tools for generic video. Extracting content from video remains one of the major problems in computer vision and current best efforts for segmenting real objects work only in limited domains with good lighting conditions. One contrived case in the broadcast world is the *blue screen* technique where anything in front of a single known colour background is considered the content of interest. In a more likely scenario of a scene with perhaps several objects of interest against a cluttered background the problem is very different. As Beaumont [11] argues, the nature of this problem depends on our application. For further compression for a videophone application a successful segmentation would have objects chosen to maximise image quality for a given bandwidth. These might include faces, hair, clothing etc., even eyes and mouths but perhaps not static background or static parts of the person. For virtual reality applications, segmented objects would have to be similar to our own subjective appreciation of complete objects or otherwise the illusion would fail. These are conflicting object definitions, but both can be reconciled as appropriate collections of content sub-regions. The content definition problem then becomes one of defining appropriate techniques to build these sub-regions and appropriate rules of combining them to obtain our content of interest.

A number of approaches have been made in the area of object segmentation and content definition by the computer vision and video coding research communities. These include histogram thresholding, changed area detection, colour clustering, region growing, eigen-value decomposition, motion segmentation and morphological processing amongst others and this thesis examines developments in this area by the author and others.

1.3 Thesis Overview

Both motivations and application domains are reflected in the organisation of this thesis. Further compression research work was centred around the development of a simulation model for object based coding known as SIMOC and also the development of a near-term low bitrate coding standard for real-time communication known as ITU-T Recommendation H.263. Both these activities were carried on in a collaborative effort within a European project known as COST 211 and also the ITU-T SG 15. A review of compression techniques, including standardised coding techniques and algorithms, model-based coding, layered coding and object-based coding is given in chapter 2. Chapter 3 looks at the development of object-based coding within a common simulation model known as SIMOC and extensions to the basic model, particularly with regard to motion analysis and motion modelling. Chapter 4 details the ITU-T Recommendation H.263 which was developed in parallel with SIMOC but built on existing block based techniques. Research into extensions to H.263 is also detailed including intra block prediction and incorporation of better entropy coding.

Video content definition research was carried out within the COST 211 project and the MPEG-4 standardisation activities. Chapter 5 introduces ISO/IEC MPEG-4, the first standardised algorithm with content-based functionality at its core, which, although it would not specify content analysis techniques for generic video, would benefit significantly from them. Chapter 6 details research into one analysis technique known as feature clustering which aimed to build pixel feature vectors made up of multiple image cues such as colour, texture and motion and combine in an appropriate feature space clustering algorithm. This chapter also examines the use of this approach in tracking of content through whole sequences. Chapter 7 details the current state of the COST 211^{quat} Analysis Model, which represents a concerted European effort to combine techniques of content definition and tracking specifically aimed at taking advantage of developments in the standards arena. This chapter also details and compares other approaches and hybrids such as Recursive Shortest Spanning Tree (RSST), morphological processing and the watershed algorithm, hierarchical pyramid region growing and the author's feature clustering work. Segmentation frameworks from the standards community are also introduced. After the conclusions in chapter 8, a number of published papers are appended.

2. Compression Techniques Review

This chapter looks at the background to the compression part of this research work. Section 2.1 details the tried and tested techniques used within the digital video standards community and their combination in the ITU-T H-series Recommendations and ISO/MPEG standards. Section 2.2 looks at model based compression for videophone application using three dimensional wire-frame models of a human head and shoulders. Sections 2.3 and 2.4 look at more generic modelling ideas that formed the basis of further compression research.

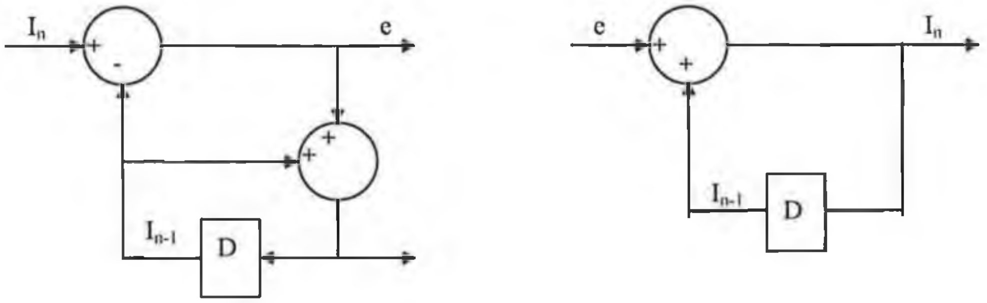
2.1 Standardised Techniques

A combination of statistical redundancy reduction techniques and perceptual redundancy techniques are employed in the modern video compression standards and this section details the main techniques of each and the algorithmic details of H.261, H.263, MPEG-1 and MPEG-2.

2.1.1 Statistical Redundancy Reduction

2.1.1.1 Predictive Coding

Predictive coding allows information about pixels already transmitted to a decoder to be used as a prediction for subsequent pixels not yet sent and exploits the fact that temporal and spatial correlation in sequences of natural images is very high. An illustration of a basic predictive encoding and decoding loop is given in Fig. 2.1



a) Encoder *b) Decoder*

Figure 2.1 : Predictive Coding Loop

Here, I_n represents the current image pixel intensity, I_{n-1} represents a previously transmitted pixel value and D is a variable delay store. Depending on the value of this delay we could use the pixel just transmitted (spatially to the left), the pixel one raster line previous (spatially above) or the pixel one frame previous (temporally before). If spatially derived predictors are used the process is termed *intra-frame* prediction and if the predictor is temporally derived, the term is *inter-frame* prediction. The probability density function (PDF) of the prediction error signal, *e*, for inter- or intra-frame case, is strongly peaked around zero and entropy coding techniques, see later, can be effectively employed to achieve significant compression.

2.1.1.2 Motion Compensation

The inter-frame predictive coding technique described only works well in stationary parts of the sequence. In moving areas the best predictor may not be the one exactly one frame behind but it may be *x* pixels to the right and *y* pixels above in the previous frame. This offset *{x,y}* is termed a *motion vector* and can be included in the coded bit stream. The overhead associated with transmitting a motion vector for every pixel is too high, so standard techniques employ a compromise of one motion vector for a square area of 8x8 or 16x16 pixels. This enforces a restrictive model of purely translational block motion that does not support typical image deformations, rotation, zoom etc., but is effective in reducing prediction error signal entropy.

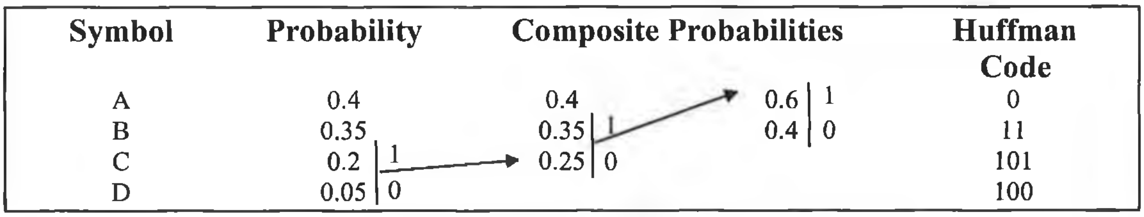
2.1.1.3 Entropy Coding

Any digital information source that outputs symbols from a fixed finite alphabet with different statistically determined symbol probabilities is amenable to compression, and the more skewed the probabilities, the more compressible the data stream. The more frequently occurring symbols, for example zero in the prediction error signal described earlier, can be encoded with less bits than those appearing less often. The *entropy* of the data stream is a measure of the average number of bits per symbol required in the optimal compression case and *variable length coding* is the term used to describe the assignment operation of different size codes to each symbol. The optimal length of a codeword for a symbol of probability *p* is given by Eq. 2.1. [2,12].

Eq. 2.1

$$l_{opt} = \log_2\left(\frac{1}{p}\right)$$

Huffman [12] devised an algorithm to optimally assign integer length codewords to individual events and an example of Huffman code generation is given in figure 2.2. These codes also have the property that they do not form a prefix to any other so each can be uniquely decoded.



Symbol	Probability	Composite Probabilities		Huffman Code
A	0.4	0.4	0.6 | 1	0
B	0.35	0.35 | 1	0.4 | 0	11
C	0.2 | 1	0.25 | 0		101
D	0.05 | 0			100

Figure 2.2 : Huffman code table generation

Codes are generated by first compiling and ordering a list of probabilities of each symbol occurring in the sample. The second stage in the process is to combine the two lowest probability messages into a composite auxiliary message of their combined probability. The combined messages are removed from the probability list and the composite probability inserted. Each combination of two messages is accompanied by the assigning of a new digit to each (1 or 0) and the step is repeated as shown until only two probabilities remain. The codes can then be read off the table by traversing from right to left (e.g. for symbol D, the composite probabilities are 0.6, 0.25 and the original probability is 0.05. The assigned code is then 100).

2.1.2 Perceptual Redundancy Reduction

2.1.2.1 Sub-sampling

The first technique we can employ here for data reduction is to reduce the amount of input information fed to our coder by spatial and temporal filtering and sub-sampling. Also, in the YC_rC_b colour space, we can differentiate between the luminance and colour difference source information and exploit the fact that human eyes are more sensitive to grey level than colour. Typically the resolution of the colour difference components can be further reduced by a factor of 2 without a perceived loss in quality. Table 2.1 specifies typical source resolutions for video compression codec standards.

Table 2.1 : Digital video source resolutions in common use.

Source	(Y)	(Cr)	(Cb)	Frame rate Fps	Pixel rate M pixels/s
BT.601 / 50Hz	720 x 576	360 x 576	360 x 576	25	20.7
BT.601 / 60Hz	720 x 480	360 x 480	360 x 480	30	20.7
CIF	360 x 288	176 x 144	176 x 144	30	4.66
SIF/625	360 x 288	176 x 144	176 x 144	25	3.88
QCIF	176 x 144	88 x 72	88 x 72	30	1.11

The gain in compression is not linear here, however, as the remaining samples usually have less correlation and the statistical techniques are less effective.

2.1.2.2 Quantisation

Quantisation is the process whereby sample values of any continuous signal are mapped to one of a fixed number of values (e.g. luminance samples mapped to range 0 to 255). The number of sample values determines the number of bits used to encode any one of them. For source video data, Recommendation BT.601 specifies 8 bits each for Y, C_r and C_b and this cannot be reduced further without introducing visible distortion. However, for difference image data, used in the predictive coding loop described earlier, the error signal is usually largest in areas of moving edges or high detail and we can effectively employ further quantisation. As the eye has reduced sensitivity to errors in these high detail regions, a psycho-visual phenomenon known as *spatial masking*, we can employ a non-linear quantiser and encode large errors with larger quantiser steps, hence fewer bits, than small ones.

2.1.2.3 Transform Coding

Image data is generally highly correlated information with low spatial frequencies. At edges and at regions of high detail spatial frequencies are higher but we can assume that this accounts for a small percentage of the picture and also that our eye has lower sensitivity to high spatial frequencies. Mathematical transformation techniques and quantisation can be used to concentrate most of the energy of a block of image data to the lower spatial frequencies. There are a number of mathematical transforms that we can use but the most widely used in image compression is the *Discrete Cosine Transform*, a variant of the *Fourier Transform* for real (as opposed to complex) data. Eq.s 2.2 and 2.3 list the forward and inverse DCT equations respectively

Eq. 2.2 $$S_{uv} = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} S_{xy}\cos\frac{(2x+1)u\pi}{16} \times \cos\frac{(2y+1)v\pi}{16} \quad (0 \le u,v \le 7)$$

Eq. 2.3 $$S_{xy} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)S_{uv}\cos\frac{(2x+1)u\pi}{16} \times \cos\frac{(2y+1)v\pi}{16} \quad (0 \le x,y \le 7)$$

$C(0) = 1/\sqrt{2}$, $C(i) = 1$ $(1 \le i \le 7)$

S_{uv} are the DCT coefficients, S_{xy} are the spatial pel values

Application of eq. 2.2 to an 8x8 block of image data transforms 64 spatial data points to 64 frequency coefficients of DCT basis functions shown below in fig. 2.3. Every image data block can be expressed as a linear combination of these basis functions.

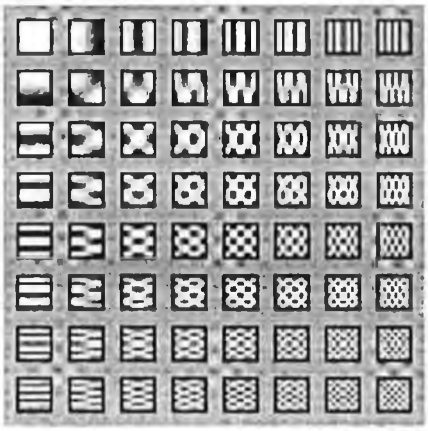


Figure 2.3 : DCT Basis Functions

Each coefficient can then be quantised differently with values stored in a psycho-visual quantisation matrix. Scaling of this quantisation matrix allows control of the amount of information to lose here and trades picture quality directly with compression ratio. The typical result at this stage is a DC-coefficient in the (0,0) position (representing the average value of the block) followed by a small number of non-zero AC coefficients clustered around the top left corner of the block (the lowest spatial frequencies). By ordering the coefficients in the *zig-zag* scan of fig. 2.4 we can compile the shortest list of non-zero terms and follow with an end of block (EOB) symbol.

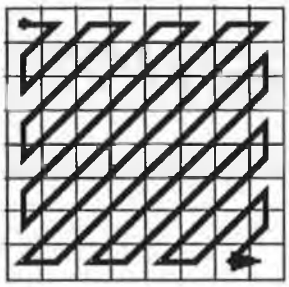


Figure 2.4 : Coefficient zig-zag scan

2.1.3 ITU-T and ISO Coding Standard Series

All these techniques were first combined in ITU-T Recommendation H.261 [13] and fig. 2.5 below shows the main processing steps of the algorithm.

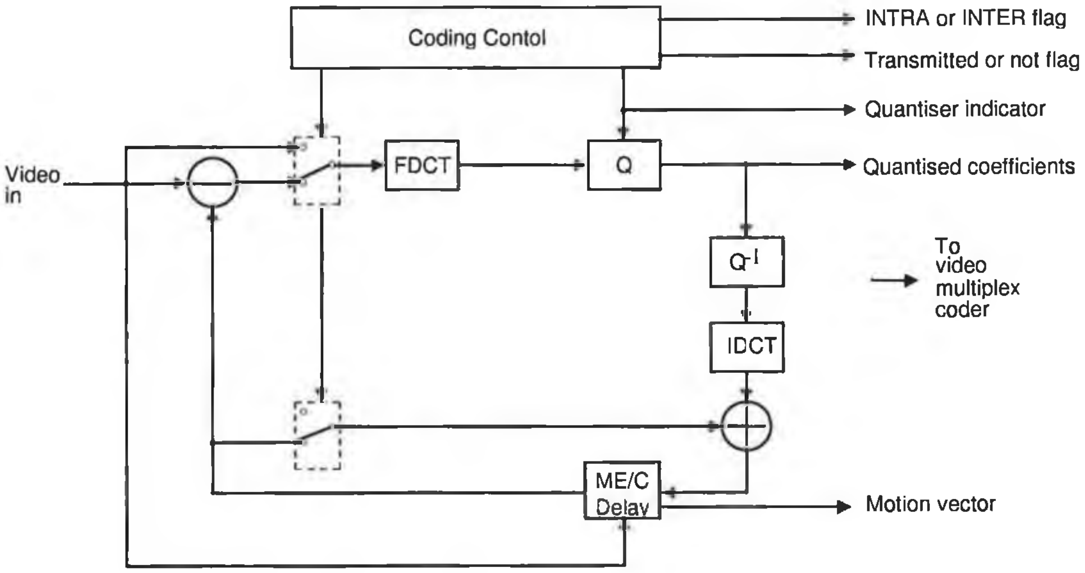


Figure 2.5: Source coding architecture of H.261

The DPCM loop is evident here with the addition of the transform and quantiser stages. Video input data is partitioned into so-called *macroblocks* made up of four 8x8 luminance blocks and two 8x8 colour difference data blocks of the same image area. The coding control block decides both whether to code as an inter or an intra macroblock and the quantiser step sizes to reach the target bitrate. The H.261 algorithm was optimised for use on the Integrated Services Digital Network (ISDN) where channel capacity of the basic unit was 64 kbit/s. The algorithm is actually specified at a bit rate range of 64 kbit/s up to 2 Mbit/s coding CIF and QCIF resolution source video.

MPEG-1 [14] was an extension of H.261 specifically aimed at the digital storage media CD-ROM market and aiming at a bit rate of 1.5 Mbit/s. Here, real-time constraints of H.261 could be relaxed and new picture types were introduced as well as support for higher resolution motion vectors for improved prediction. The inter-frame prediction coding in H.261 used only one reference frame, the previously transmitted intra-predicted picture or *I-frame* as it is termed. The inter-predicted picture is known as the *P-frame*. MPEG introduced a third picture type known as the bi-directionally predicted picture or *B-frame* which could use both a previously transmitted I-frame and a P-frame representing a picture later in time. This feat of time travel is achieved by storing up a set of frames in a buffer, coding the latest frame in the normal way as a P-frame, then returning to code the remaining buffered pictures as B-frames. The decoder must also buffer and re-order these frames before display. This allows detail not present in the I-frame due to motion (e.g. occluded information) to be included in the prediction but introduces buffering delays and more complex processing requirements. Figure 2.6 show the temporal relationship between these different picture types.

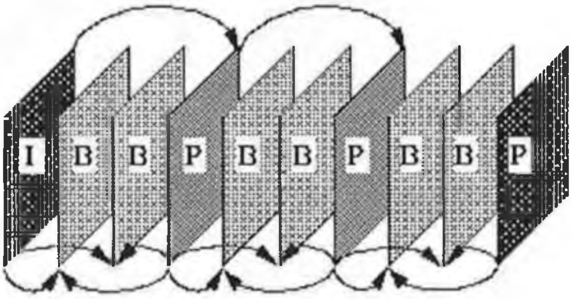


Figure 2.6 : Order of coding and prediction paths of I, P and B frames

MPEG-2 addressed higher quality broadcast video applications with added support for interlaced source formats and scalable coding. Interlaced video is made up of two fields for every frame and is a hangover from analogue television days where video of 25 to 30 frames per second was adequate for proper motion rendition but displays of at least 50 to 60 pictures were needed to avoid annoying flicker. These fields are captured at different times and line-interleaved in each frame so prediction with frame block data now makes less sense. MPEG added support for block motion prediction within fields. Scalability was the second new feature introduced, allowing one video bitstream to be partitioned into sub-streams that could be later combined to give different quality decoded video. This would mean that a standard bandwidth link could use a *base layer* of video for say TV resolution but higher bandwidth broadcast studio links could also combine the base layer with one or more *enhancement layers* to yield HDTV quality. This feature could also be used to dynamically vary streaming video quality on the internet or local intranet depending on network activity.

H.263 [15], from the outset, tackled video conferencing within the lower bandwidth range of 8 to 64 kbit/s. The recommendation was driven by progress in the modem standards (V.34) of full duplex communications at 28.8 kbit/s, and later 33.4 kbit/s, and half duplex at 56.8 kbit/s on the more widely available public switched telephony network (PSTN). The architecture of H.263 is the same as H.261 with the optimisation of certain stages and the addition of higher complexity options for enhanced prediction. Some of these enhancements were the subject of research work described later in this thesis. The main ones are: an optimised Huffman variable length coding table, higher resolution motion vectors, optional addition of more efficient motion compensation, arithmetic entropy coding and *PB-frames* (yet another picture type combining advantages of MPEG B-frames with the low delay of standard P-frames). The combination of all these enhancements delivers almost twice the compression of H.261 with perhaps a 50% increase in implementation complexity if all options are used.

2.2 Model-Based Coding

This section describes one of the earliest efforts to apply model-based video coding to videophone communication in particular. The work uses a very specific 3D polygonal wireframe model adapted to match a subject's head and shoulders and with texture from a single image of the subject mapped onto the model surface. Head movement and facial expressions in this scenario are modelled as global and local deformations of the adapted wireframe and these can be tracked at the encoder and used to synthesise images at the decoder.

2.2.1 Wireframe Model

The system described here was targeted at real time operation and consequently a low-complexity facial wireframe model, the *Candide* [5] wireframe, figure 2.7, was adopted at the outset. This initial model and the global wireframe conformation method described in [4] required fitting of just 6 control points from the overlayed mask to the facial image. Further confirmation techniques were developed, see next section, and modifications made to the basic model, shown in figure 2.8, including extra triangles corresponding to the hair, neck and shoulders and a second model for the interior of the mouth.

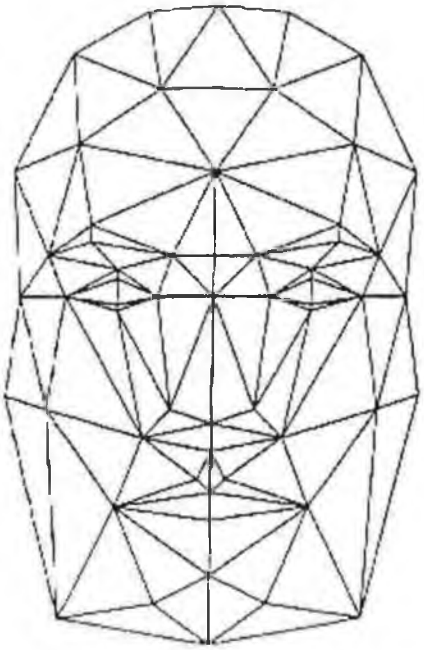


Figure 2.7 : Candide wireframe model (frontal projection)

An array of vertex x, y and z positions with vertex labelling is all that is needed to define these models.

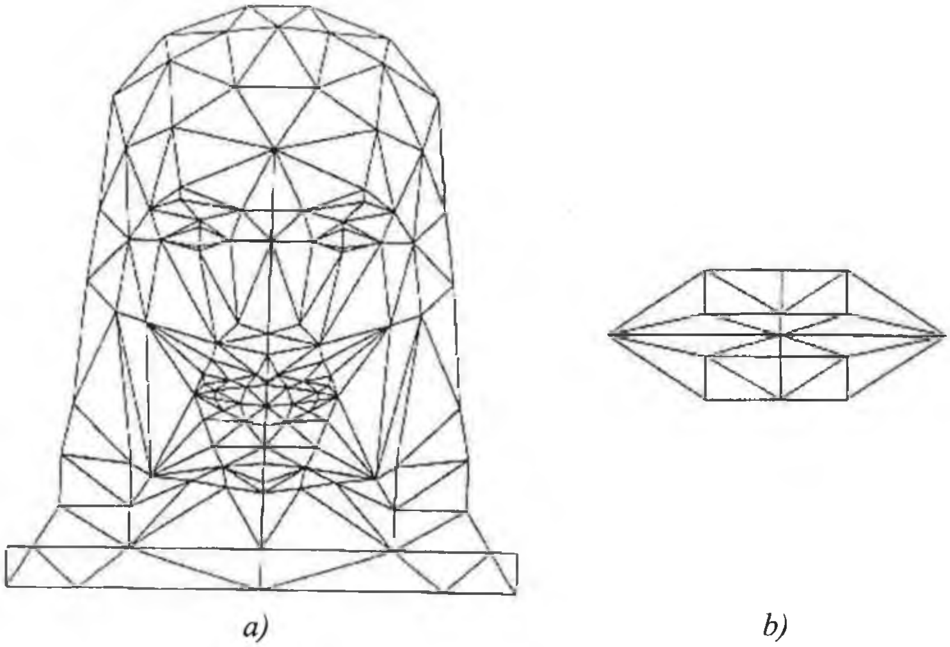


Figure 2.8 : Modified Candide a) frontal projection b) mouth interior

2.2.2 Model Conformation

Both manual and automatic model conformation were used in the work described with manual efforts proving much more successful. Automatic conformation techniques included application of Nagao [16] operators to a static head and shoulders image and the use of a fixed end snake to collapse onto the head boundary. Here the snake criterion for locating boundary fragments was not robust enough and model conformation failures were reported.

Manual techniques involved a selection of a set of control vertices within the wire-frame and manipulation of these points when overlaid on the static image contained within a GUI application. Local conformation methods with piecewise linear mapping were found to perform best here with positions of non-control model vertices influenced only by the changes in position of the control points closest to them. Two techniques, *Delaunay* triangulation between control points and linear interpolation of non-control points, were used to produce a smooth deformation of the wire-frame.

2.2.3 Image Synthesis

The illusion of realism in the synthetic images was achieved by texture mapping from an original picture onto the frontal projection of the wire-frame after motion was applied. Two types of motion handled are global motion of the wire-frame and local motion due to changes in facial expression. For both, the generation of the synthetic image is performed in two stages: first, the wire-frame is rotated and required facial expressions are applied using the Facial ACtion unitS (FACS) [17] system, second, texture is applied to the deformed wire-frame triangles. Two examples of resulting synthetic images are given in fig. 2.9.

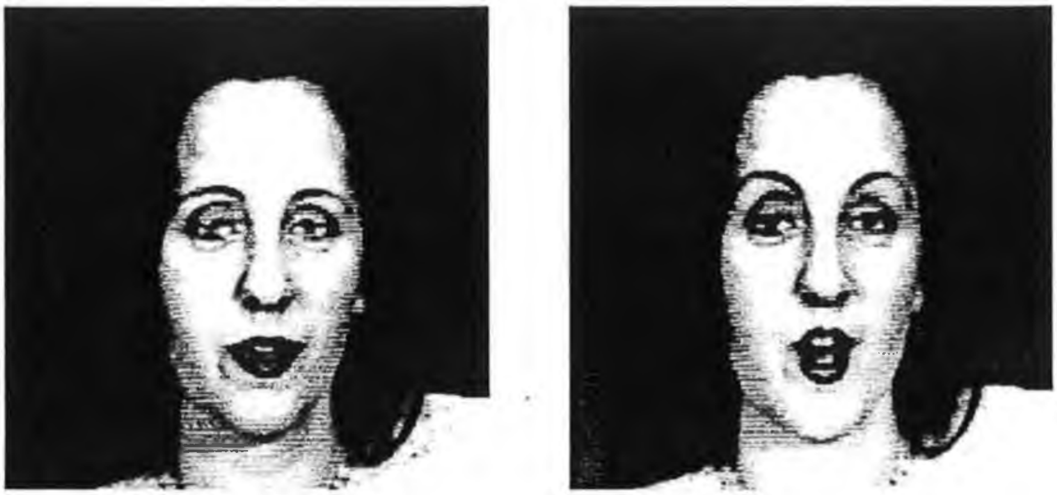


Figure 2.9 Sample synthesised (texture mapped wire-frame) images

2.3 Layered Coding

The model based coding approach is restricted to the chosen image model (e.g. head-and-shoulders type sequences in the scheme described). The layered coding [7] approach from MIT Media Lab is, on the other hand, applicable to a wider range of image material. Their technique is based on the exploitation of mid-level vision concepts - specifically coherent motion surfaces and image regions whose properties are "considered sophisticated enough to produce powerful representations yet simple enough to be computed". Their approach uses these vision techniques to derive a representation of the video content in terms of overlapping layers. Each layer represents content data (colour, intensity, transparency and motion) at different depths in the scene and once obtained can be used to re-compose the original scene by direct layered composition.

The challenging problem here, as in model based coding, is the initial decomposition or analysis problem. The MIT approach is based on motion analysis of the sequence. The first stage is the calculation of a dense motion vector field using optic flow (assuming translational motion) at every pixel, from which an affine model of motion is derived. Segmentation is then achieved by assigning each pixel location to one of the derived affine models. Where coherent motion regions are tracked over many frames a single layer intensity map can be deduced.

The coding representation achieves enhanced data compression by encoding each layer component map separately. The intensity and colour information is coded with the DCT with modifications to edge blocks to enhance energy compaction of the DCT process. Shape information, in the form of an alpha transparency map is encoded using a chain code contour algorithm.

Encouraging results are presented in [7], which show the effectiveness of this coding approach on the MPEG *Flower garden* sequence. This sequence contains a pan with a row of houses in the background, a tree very close in the foreground and a flower garden in between. A very clear layered motion is evident here with layers of content moving with different translational speed depending on the depth and the technique works very well. The content is still quite restricted, however, to video which can be decomposed in this way.

2.4 Object Based Coding

The previous sections represent the extremes of content based coding at the outset of this research work. The standardised techniques treat video as a sequence of 8x8 block grids. Model Based coding, on the other hand, presumes a very specific type of content only really suitable for one-to-one video conferencing. Layered coding is more generic, but still relies on sequences that can be adequately described in terms of layered motion. This section discusses in particular the even more generic model based coding approach known as *Object-Based Analysis/Synthesis Coding (OBASC)* based on the University of Hannover's work [8].

2.4.1 Object-Based Analysis/ Synthesis Coding

The technique described here is based on a source model of moving 2D objects where images are sub-divided into moving objects and each object is described by three sets of parameters for motion, shape and colour information. The parameter sets are stored in parameter memory both at receiver and transmitter and used to reconstruct the transmitted picture by image synthesis techniques. Figure 2.10 shows an OBASC encoder and the associated parameter memories.

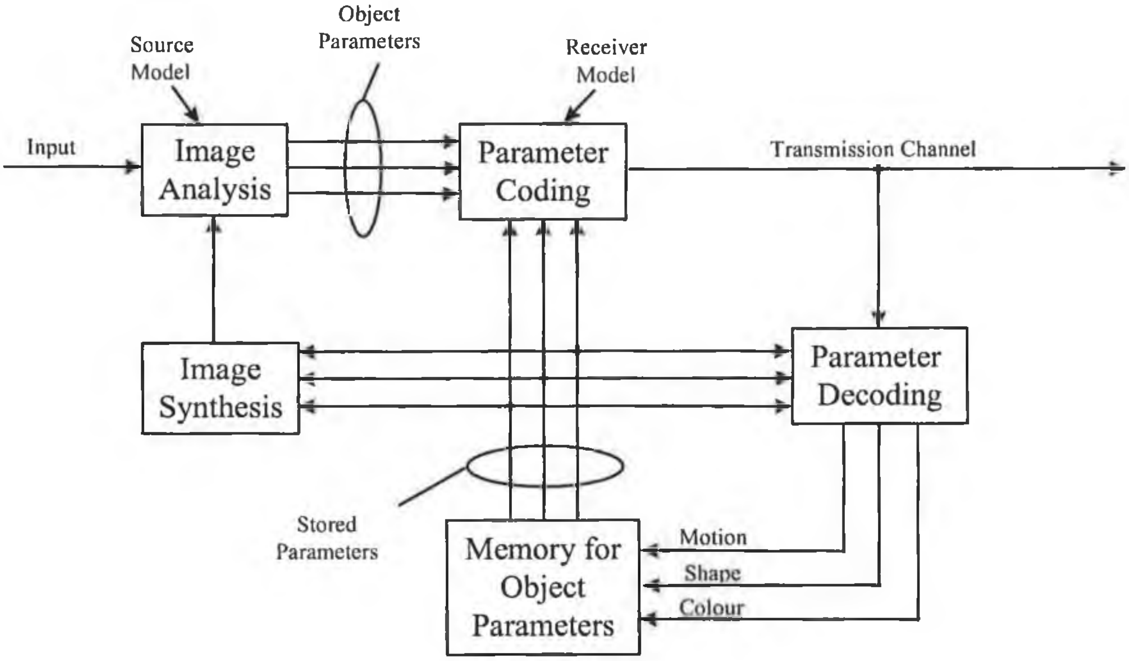


Figure 2.10 : Block diagram of object-based analysis-synthesis coder.

Three sets of data are transmitted in this coding scheme (colour, motion and shape) rather than two for the standard coders. To achieve compression gain, the additional bits required for transmitting shape information must be more than compensated for by a reduction in bits required for motion and colour. Early work with this scheme on test sequences such as *Miss America* and *Claire* showed that this was indeed the case and the European COST 211ter group developed the technique further in a simulation model for object based coding known as SIMOC [19].

2.4.2 **SIM**ulation model for **Ob**ject based **C**oding (SIMOC)

The main algorithmic stages of SIMOC are as follows:

- The first frame of the sequence is coded as in standard block coders with the intra coding techniques described earlier.
- Differences between the last predicted frame and the current one are also detected as before. However, now, a frame sized binary *change detection mask* is constructed defining static and changed pels.

- Motion is analysed only for segments that differ from the last frame and estimated motion vectors are used to distinguish moving areas and uncovered background areas. A second ternary mask of **S**tatic, **M**oving and **U**ncovered background, (*SMU mask)*, is also constructed at this stage.
- A synthesised frame can now be generated with the previous reconstructed frame, the estimated motion vectors and the SMU mask. Within the moving areas, regions predicted well are termed *Model Compliant (MC)* objects and those not so well predicted are termed *Model Failure (MF)* objects. MF objects and uncovered background texture must be coded using arbitrary shape VQ or shape adaptive DCT techniques.

2.4.2.1 Change Detection

The change detection mask is constructed by the following steps:

- Absolute differences between the current and previous reconstructed frames are computed.
- For each pel in the luminance difference image, the 3x3 window neighbourhood centred on that pel is summed. A neighbourhood sum value greater than a fixed threshold defines that pel as *changed.* Below this value the position is marked as *unchanged.*
- A 5x5 median filter is applied to the resulting changed and unchanged regions.
- Moving areas from previous SMU masks *(*masks prior to shape approximation and retained in parameter memory store) may be added and also marked as changed. This step enforces some stability of the object mask where portions of the object stop moving but must be controlled to avoid changed area accumulation.

The mask is typically noisy at this stage. Dilation and erosion morphological operations with a 3x3 structure element and a final elimination operation of small sized areas are also carried out to yield the final mask. Figure 2.11 shows a typical result of change detection near the start of the *Mother and Daughter* test sequence. Change detection threshold was set to 18 and scene is from first few seconds of the sequence.

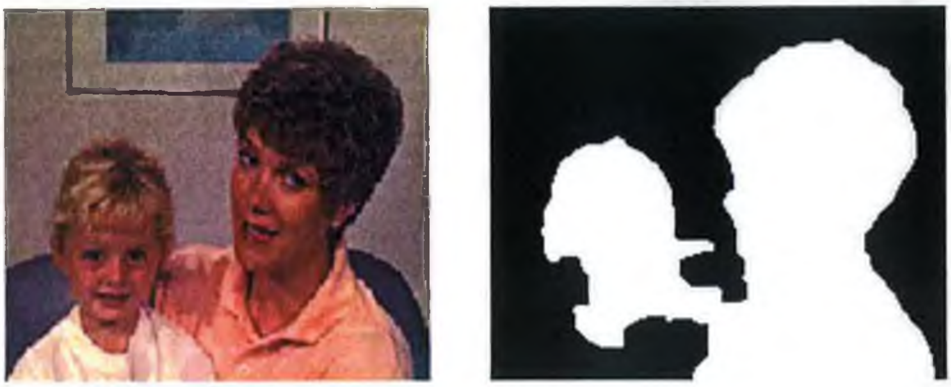


Figure 2.11 : a) Scene from Mother and Daughter and b) Change Detection mask

2.4.2.2 Motion Analysis & SMU Segmentation

The next stages of the SIMOC algorithm require the generation of the ternary Static, Moving and Uncovered (SMU) mask and the detection of the model compliant and model failure regions and these in turn require motion analysis at a pel by pel level. Hierarchical block matching for calculation of grid vectors, based on Bierling's work [18], and bilinear interpolation between these vectors are used to calculate a motion vector for each changed pel. Motion analysis is one area where improvements were made to this algorithm by the author's research and is described in chapter 3. These pel motion vectors are then used to generate the SMU mask. The Change Detection (CD) mask is first used to initialise the S and M parts of the ternary mask. 'Changed' pels as determined from the CD mask may be changed due to having moved or the may be uncovered background. As each 'changed' pel also has an associated motion vector this can be used to determine which category the pel is in by comparing the pel itself and the pel in the previous frame that the vector implies is the source. If these pels are sufficiently close the pel is probably moving area. If not the pel is most likely to be uncovered background. The SMU mask bits can now be set accordingly.

2.4.2.3 Model Compliance / Model Failure Decision

All pels that are deemed to be moving can be synthesised using the previously reconstructed frame and the motion vectors. These synthesised regions can then be tested against the underlying source model of flexible 2D objects. Model Failure (MF) areas are detected by establishing a synthesis error threshold for each Model Compliant (MC) object and then marking all those pels exceeding this threshold as MF regions. The threshold T, for each MF object is calculated using the following procedure from [19]:

1. Set $T_{MF}=1$
2. Calculate the synthesis error variance, MSE_{syn}, for all pels of the object whose synthesis error is less than T_{MF}
3. If $MSE_{syn} < 6$ set $T_{MF} = T_{MF} + 1$ and go to step 2. Otherwise set $T_{MF} = T_{MF} - 1$ and finish.

As with the change detection mask, the resulting binary MF region mask is median filtered, dilated and eroded and small areas eliminated. All non-MF regions completely surrounded by MF regions are also eliminated. The region is finally delineated by a chain code boundary definition algorithm.

2.4.2.4 Shape Approximation

A polygon approximation method is used to approximate MC and MF object shapes for subsequent coding. The following steps, and fig. 2.12, define the process.

- Obtain the main axis of the object. The points of maximum separation on the object contour define the first two vertex points 1 and 2.
- Obtain remaining initial vertices. These are the points of maximum perpendicular distance to the main axis 3 and 4.

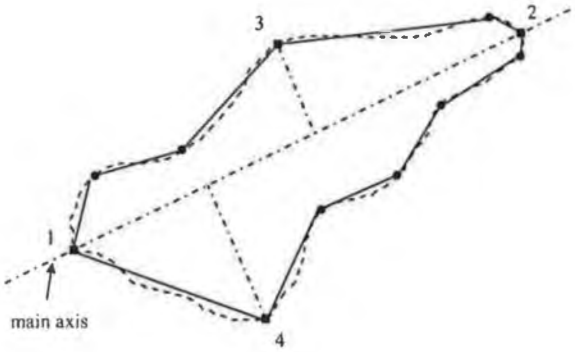


Figure 2.12 : Polygon approximation of MC and MF objects

- The procedure described applies to MF objects and MC objects with no reference in the previous frame i.e. newly determined objects. For MC objects with a reference in the previous frame the previous vertices may be displaced with the estimated motion and the closest new contour points used in subsequent approximation steps.
- Complete polygon approximation. This is achieved by drawing straight lines between adjacent vertices and ensuring that any contour points are no more than 2

pels perpendicular distance outside, or 1 pel distance inside, the object from these lines. When the entire contour has been processed the list of vertices forms the polygon approximation.

2.4.2.5 Parameter Coding

The analysis stage of SIMOC produces three sets of parameters for motion, shape and colour.

- Motion parameters take the form of a grid of displacement vectors representing translational block motion only. SIMOC uses vectors in the range +/- 4½ pels on a 16x16 grid. Delta vector and adaptive arithmetic coding is used for the motion field entropy coding.
- Shape coding requires the description of the x and y vertex positions relative to the nearest neighbour (counter clockwise). 2-dimensional events for all objects are coded again by an adaptive arithmetic coder.
- Colour information must be transmitted for MF regions and uncovered background prediction errors. Spatial vector quantisation (VQ) is used here. Vectors consist of the 4 data points within the 2x2 luminance image blocks and corresponding 2 data points of the colour difference data giving 6 component vectors. The positions of all these blocks is given by a simple tessellation of the image and only blocks that lie fully within the approximated MF region are coded. The PSNR threshold for VQ codebook candidates is set at 31dB and the codebook indices are coded once again by adaptive arithmetic coding.

2.5 Review Summary

Video coding has enabled many applications including video-telephony, video-on-demand, personal computer based multimedia and tele-conferencing. The successful market development of many of these areas has been absolutely dependent upon agreed standards, principally those from the ITU-T SG 15 and ISO/IEC MPEG. These standards in turn have been the fruits of a number of years of research into well understood statistical and perceptual redundancy reduction techniques. In terms of compression, however, these techniques appear to be reaching an asymptote. Model-based coding, Layered coding and Object based coding approaches, on the other hand, are new coding techniques utilising much more *a priori* knowledge about scene content to push compression further. All these new approaches are changing the general model of video from static square blocks with *x* and *y* translational motion to either very specific head and shoulder models or more generic overlapping layers or arbitrary shape regions with more complex affine motion models. These more complex video models do imply more complex computational requirements and it remains to be seen where the correct balance is between achievable quality and codec complexity. Both extensions to the SIMOC object-based coding model and the H.261 standardised model are investigated in the following chapters.

3. Motion Estimation for Object Based Coding

3.1 Introduction

The object-based analysis-synthesis coding scheme, proposed in SIMOC-1 [19], represented the most promising and generally applicable knowledge based coding approach at the outset of this research and provided a good basis for further research into content based coding. SIMOC-1 was proposed by the COST 211ter [20] video coding group, the same group that proposed what later became H.261 and MPEG-1, and a number of algorithmic features (e.g. motion analysis) are common between these algorithms. Motion estimation within the standards compliant encoders is commonly performed on a block basis to keep computational load low and to avoid any requirement to spend bitrate on shape coding. Most estimators also aim to minimise displaced frame difference to predict the lowest entropy error signal before interframe coding. The resulting motion vector field grids are then a compromise between true motion representation, motion estimation complexity and bit rate.

SIMOC uses a hierarchical coarse to fine estimator, described in section 3.3, both to reduce computation and supply a generally robust vector field. However, the motion information is transmitted as a relatively sparse 9 x 11 vector grid, with vectors representing a 16 x 16 pel block, and advantage is not fully taken of the shape segmentation and coding information. Motion within SIMOC, as opposed to traditional motion compensation coding, plays a much more direct role in the quality of the final synthesised frames and was considered an important area for further investigation. Small geometric distortions due to incorrect motion estimation are tolerated within the Model Compliant regions and only the Model Failure regions of large synthesis error are corrected without relying on the motion estimates.

Work by Niewglowski et al. [21] and others was underway at the time of this research to define much more compact parametric representations of dense motion vector fields and specifically for use within region based coding. This would allow representation of much more complicated regional real world motion such as zoom, shear and rotation without the overhead of transmitting vectors for each pel. Subsequent work [22] within MPEG-4 has shown this approach to work better, at least on a bitrate/quality criteria, than even the most highly optimised block based approach with H.263. The price paid is increased motion estimation complexity. Both H.263 and MPEG-4 are described later in this thesis.

The motion estimation research described in the next sections attempted to work with denser vector field grids than those currently used in SIMOC. The *combined displaced frame difference and Gibbs modelled vector field smoothing* technique, described by Stiller [23], was used to manage the increase in vector field entropy and bitrate of these denser grids. In section 3.2 I describe the direct application of DFD/Gibbs vector-field smoothing to 8x8 pel block and 4x4 pel block grids within SIMOC. I then look closer at the nature of the full mean absolute difference error surfaces themselves and investigate yet another smoothing proposal. Section 3.3 details the novel use of the *Hessian* operator as a local region activity and motion estimate confidence indicator for use with the smoothing technique.

3.2 Combined DFD/Gibbs modelled vector field smoothing

Motion vector fields for image coding are typically generated using block matching techniques between successive frames of an image sequence. The criteria used in such schemes is the minimum displaced frame difference (DFD) or mean absolute difference (MAD) between displaced blocks. Stiller has proposed an extension of this using both a displacement model of minimum MAD and a vector-field model favouring segment-wise smooth vectors.

3.2.1 Vector and Displacement Models.

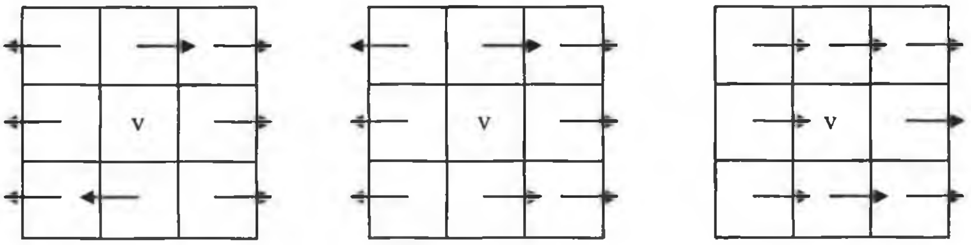
The vector interdependency is modelled by a second order Gibbs/Markov random field given by:

Eq. 3.1 $$p(V) \propto \exp(-\sum_{i=1}^{M} C_i),$$

where M denotes the number of cliques (collection of neighbourhood vectors) and C_i stands for the cost associated to the i^{th} clique. A cost-function for vectors v_1 and v_2 of a clique favouring locally smooth vectorfields can be expressed as:

Eq. 3.2 $$C = \frac{c}{l} \cdot \left\| v_1 - v_2 \right\| \text{ ; c = constant.}$$

l denotes the distance between the considered pels or blocks (1 for four nearest neighbours and √2 for diagonal ones). Figure 3.1 shows how this model behaves in different neighbourhoods.



$\Sigma C(v = \leftarrow) = \Sigma C(v = \rightarrow)$ $\Sigma C(v = \leftarrow) > \Sigma C(v = \rightarrow)$ $\Sigma C(v = \leftarrow) >> \Sigma C(v = \rightarrow)$

Fig 3.1 : Cost function for vector v *with differing contexts.*

Stiller's displacement model is derived by maximising the *a posteriori* probability density for the vectorfield given the frame pair that it relates to. The final cost function which he uses is derived in [23] to be:

$$\text{Eq 3.3} \qquad N.\mathit{ln}(\mathit{max}(MAD_v, N.P_c)) + \sum_{n=1}^{8} \frac{c}{2l}\left\|\bar{v} - \bar{v}_n\right\|$$

where *N* is the number of pels in block, MAD_v is the mean absolute difference of changed pels within displaced block, P_c power of camera noise estimated from image, *v* is the test candidate vector and v_n the neighbouring vectors.

The idea is to estimate a motion field which minimises the cost functional. This can be done iteratively by 'relaxing' the vector field parameters at each site in the image. Konrad and Dubois [88] present just such a framework. Stiller recognised that the complexity of this process could be dramatically reduced by employing only a subset of vector candidates for evaluation. This subset was chosen from the immediate neighbourhood of each site visited. The scheme worked because a simpler motion estimation process e.g. block-matching was first employed to generate 'almost correct' motion estimates at all sites. The test vector candidates are listed below:

- vector calculated by the previous estimation step
- eight neighbours calculated by previous estimation step
- four vectors differing from the first candidate by ½ pel
- weighted average of the eight neighbour vectors applying weights 1/l
- predicted vector of the predictive vector coding

3.2.2 Modifications for SIMOC

The change detection mask is available within the SIMOC-1 scheme and can be used in the vector field model of the cost function. We need only consider the neighbouring vectors if they are within this mask. This would allow discontinuities in the vector field at the mask boundaries where they should be expected to be. A second modification made was in the choice of test vectors. As the arithmetic coding used in SIMOC-1 does not use the same prediction mechanism as Stiller's coder the prediction test vector was omitted. The modified cost function used therefore is -

$$\text{Eq 3.4} \qquad N.\mathit{ln}(\mathit{max}(\text{MAD}, N.P_c)) + \sum_{n=1}^{8} M.\frac{C}{2l}\left\|\bar{v} - \bar{v}_n\right\| \rightarrow \min$$

where M=1 if v_n footpoint lies within the change detection mask, otherwise M=0, and the test vectors chosen are -

- vector calculated by the previous estimation step
- eight neighbours calculated by the previous estimation step
- four vectors differing from the first candidate by ½ pel
- weighted average of the eight neighbour vectors applying weights 1/l

The effect of smoothing itself can be seen in figs 3.2 and 3.3 where the generated field is smoother with less spurious vectors and the field entropy can be seen to be reduced. The contention is that this more closely matches the real motion in the scene while also reducing the total parameter bit rate and this is what was tested here.

Stage parameters for the 3 level HBM algorithm of SIMOC are given in Table 3.7 later in the chapter. A relatively dense field is shown in figs 3.2 and 3.3. (44x36 vectors). C was set in range ($20 \leq C \leq 80$) for fig 3.3.

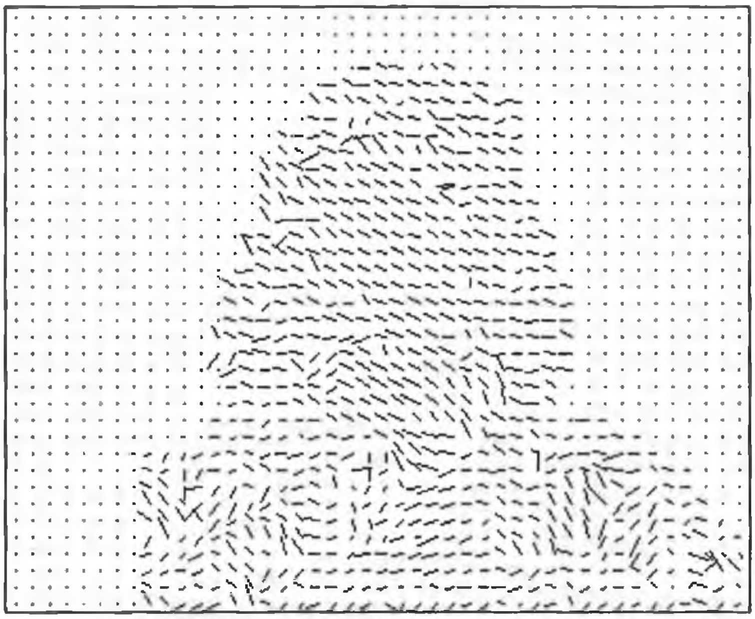


Figure 3.2 : Hierarchical Block Matching derived vectorfield for frames 75 and 78 of original Miss America sequence.

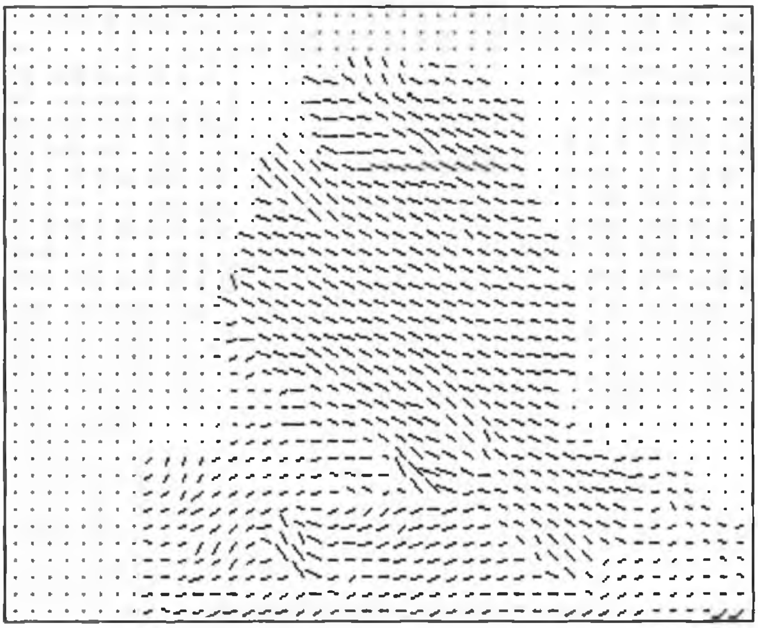


Figure 3.3 : Displaced Frame Difference/Gibbs-smoothed SIMOC derived vectorfield for identical image pair.

3.2.3 Results from SIMOC with DFD/Gibbs smoothing.

3.2.3.1 Varied Grid Sizes

The first set of results show bit counts for all motion, colour and shape parameters for the test sequence *Miss America* coded using SIMOC-1 at three different vector grid densities and three extremes of smoothing applied to the vectorfield. 50 frames of the

QCIF sequence at a framerate of 8.33Hz (i.e. 1 in 3 frames) were used. Case 1, 2 and 3 relate to no smoothing applied, smoothing with vectorfield weighting c = 20 and smoothing with weighting c = 80 respectively.

Table 3.1 - Average bit counts per frame for QCIF Miss America (11x9 vectors)

	16 x 16 Block Grid			
Case	#bits motion	#bits colour	#bits shape	#bits total
1	361	12666	434	13461
2	346	13074	443	13863
3	271	13804	414	14489

Table 3.2 - Average bit counts per frame for QCIF Miss America (22x18 vectors)

	8 x 8 Block Grid			
Case	#bits motion	#bits colour	#bits shape	#bits total
1	1538	6398	551	8487
2	1223	8085	475	9783
3	737	13055	411	14203

Table 3.3 - Average bit counts per frame for QCIF Miss America (44x36 vectors)

	4 x 4 Block Grid			
Case	#bits motion	#bits colour	#bits shape	#bits total
1	6612	1599	586	8797
2	3551	3155	575	7281
3	1586	15868	354	17808

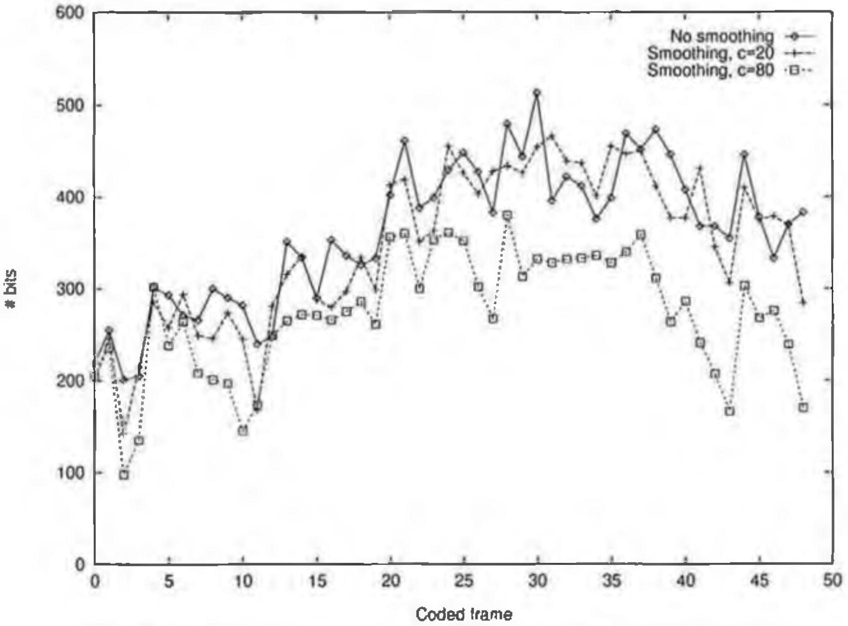


Fig 3.4 : Motion parameter bitcounts (16x16 blocks)

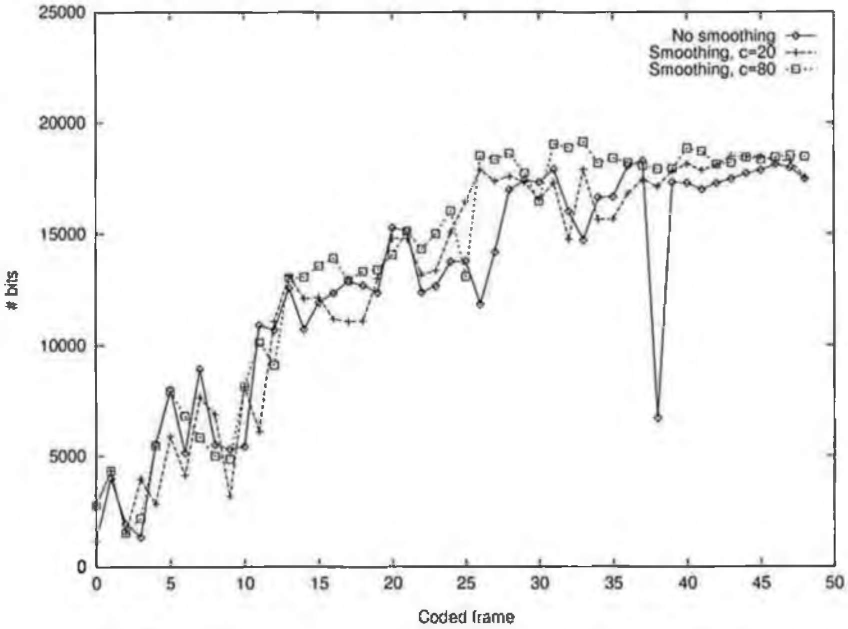


Fig 3.5 : Colour parameter bitcounts (16x16 blocks)

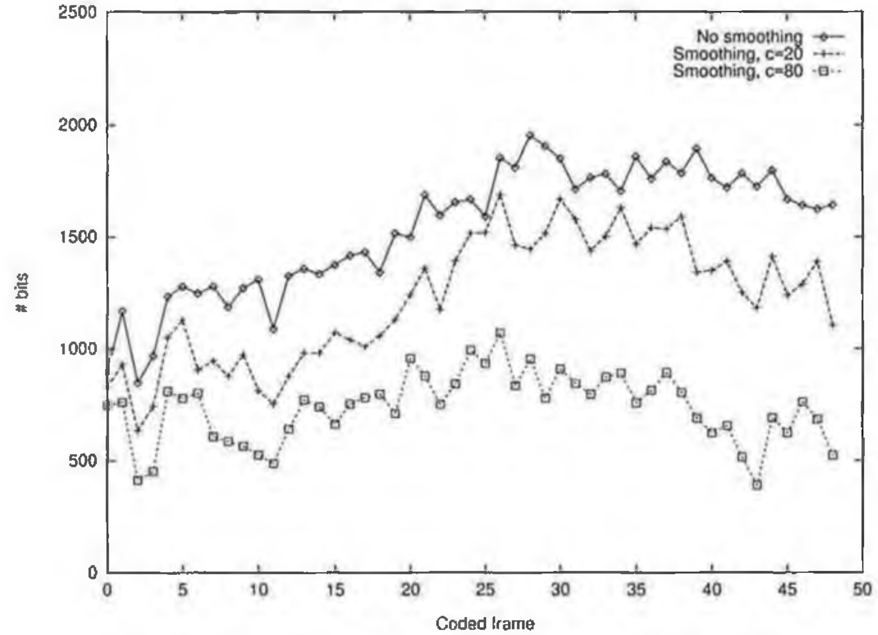


Fig 3.6 : Motion parameter bitcounts (8x8 blocks)

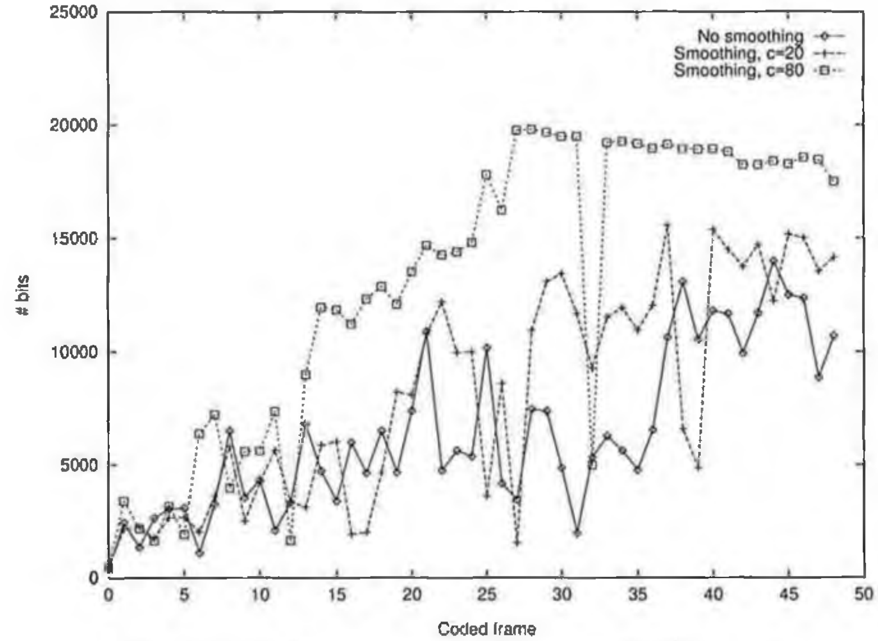


Fig 3.7 : Colour parameter bitcounts (8x8 blocks)

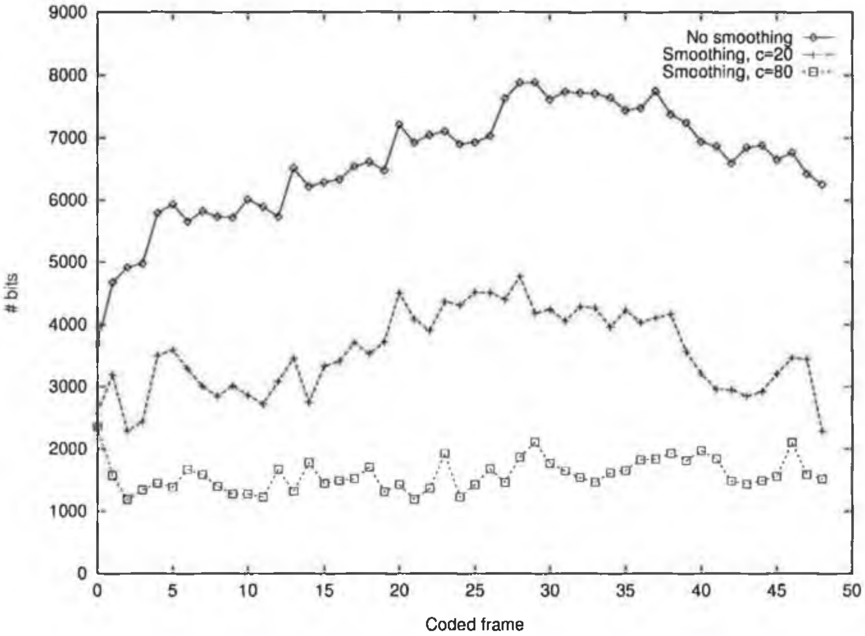


Fig 3.8 : Motion parameter bitcounts (4x4 blocks)

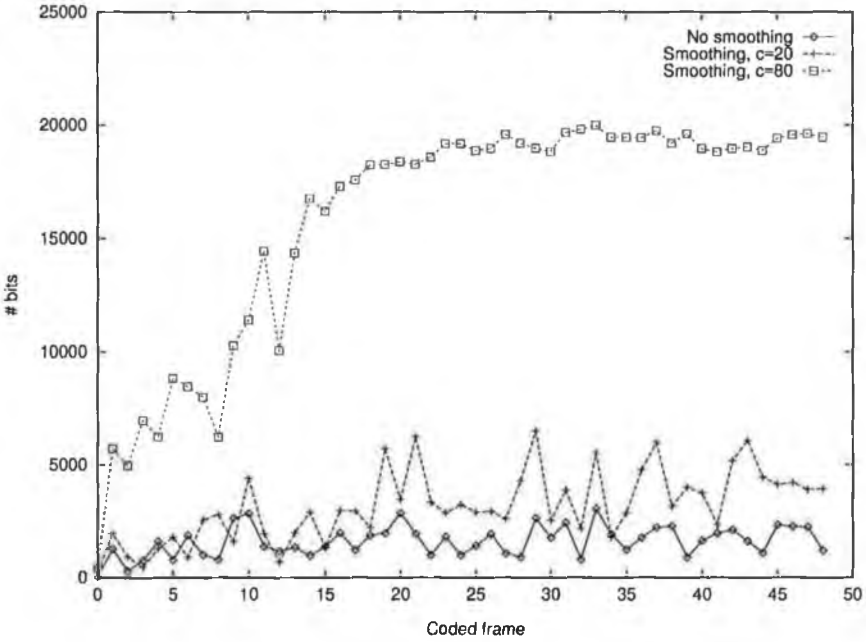


Fig 3.9 : Colour parameter bitcounts (4x4 blocks)

Tables 3.1 to 3.3 and figures 3.4 to 3.9 summarise the results for this test set. The motion parameters are coded as in the specification document [19] with a simple previous block prediction followed by adaptive arithmetic coding. The vectors themselves are calculated using the hierarchical block matching algorithm with ± 4½ pel range however the block sizes used are proportional to the grid density chosen. The colour parameters are coded using DPCM rather than vector quantisation (hence the high figures) but results should hold for VQ coding too. Some early conclusions can be made on the basis of the initial set of tests:

- The interdependency between motion and colour parameter bitcounts is strong for SIMOC with an inverse proportionality present. Shape bitcounts are largely unaffected. Relative to the specified 16x16 block grid vectors the 8x8 and 4x4 block vectors, with no smoothing applied, are more expensive bitwise for the motion parameter coding but overall much more efficient due to reduction in colour DPCM bits.
- For all grid densities the vectorfield's entropy is reduced when smoothing is used, as expected, and a lower bit rate for the motion parameters is achieved.
- Any modification of the motion vectors, certainly for a DFD full block search and probably for a hierarchical block search, requires that we move away from the minimum error condition in a frame difference sense. The energy of the resulting prediction error image can only increase by using the smoothed vectors. This increases the amount of model failure area and the colour parameter bitrate for the same grid density. It is only in the 4x4 grid case 2 results that the increase in colour bits is offset by the decrease in motion bits for the smoothed vector case.
- Although giving the best motion parameter bit rate reduction, the smoothing case with c=80 is found to be too extreme in general. In the 8x8 block grid case 3 a saving of 490 motion bits costs 4970 DPCM colour bits.

Subjectively, the sequences coded with or without smoothing showed little difference for this set of results and the luminance signal to noise (PSNR) measures are also very similar. This is due mainly to the model failure detection method, defined in [19], which 'targets' a certain overall quality; in this case an SNR of ~36dBs, and which is similar for all tests in this section.

3.2.3.2 Varied Grid Sizes with Constrained Model Failure Area

The second result set relates to a modified SIMOC encoder where the model failure mechanism was constrained to limit the final datarate. An arbitrary limit of 1000 MF pels was used, which equates to 4% of the QCIF frame data. The purpose here was twofold, firstly to reduce the bitrate to below 32 kbit/s and secondly to counteract the increase in prediction error signal due to smoothing and allow the motion information to more directly influence the quality of the output image.

Table 3.4 : Average bitcounts per frame for Miss America and luminance SNR in dBs

	16 x 16 Block Grid				
Case	#bits motion	#bits colour	#bits shape	#bits total	avg. Y-SNR
1	356	1399	562	2317	34.91
2	326	1406	571	2303	34.90
3	255	1387	513	2155	34.72

Table 3.5 : Average bitcounts per frame for Miss America and luminance SNR in dBs

	8 x 8 Block Grid				
Case	#bits motion	#bits colour	#bits shape	#bits total	avg. Y-SNR
1	1562	1414	562	3538	36.12
2	1169	1324	550	3043	35.82
3	724	1406	503	2633	34.63

Table 3.6 : Average bitcounts per frame for Miss America and luminance SNR in dBs

	4 x 4 Block Grid				
Case	#bits motion	#bits colour	#bits shape	#bits total	avg. Y-SNR
1	6634	1571	601	8806	37.47
2	3555	1450	544	5549	36.69
3	1524	2027	417	3968	30.14

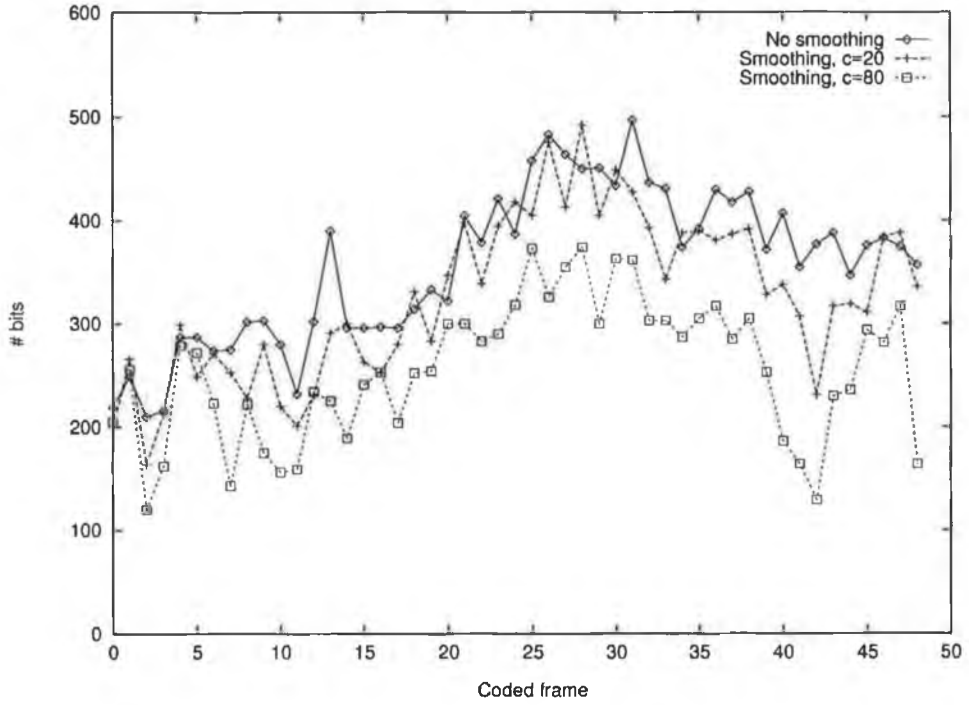


Fig 3.10 : Motion parameter bitcounts (16x16 blocks, constrained model failure area)

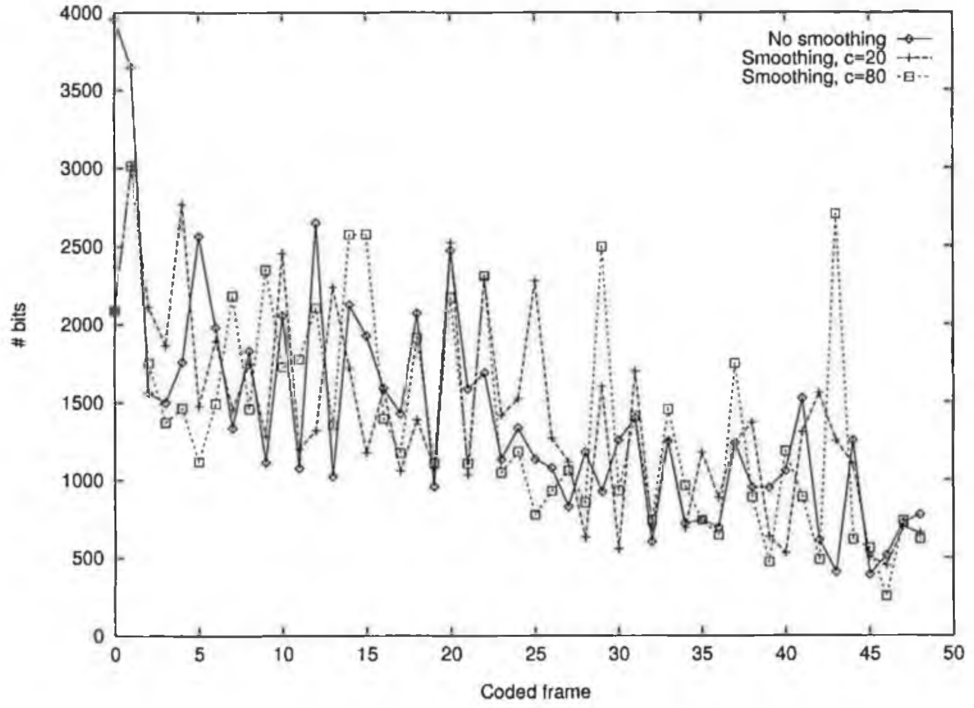


Fig 3.11 : Colour parameter bitcounts (16x16 blocks, constrained model failure area)

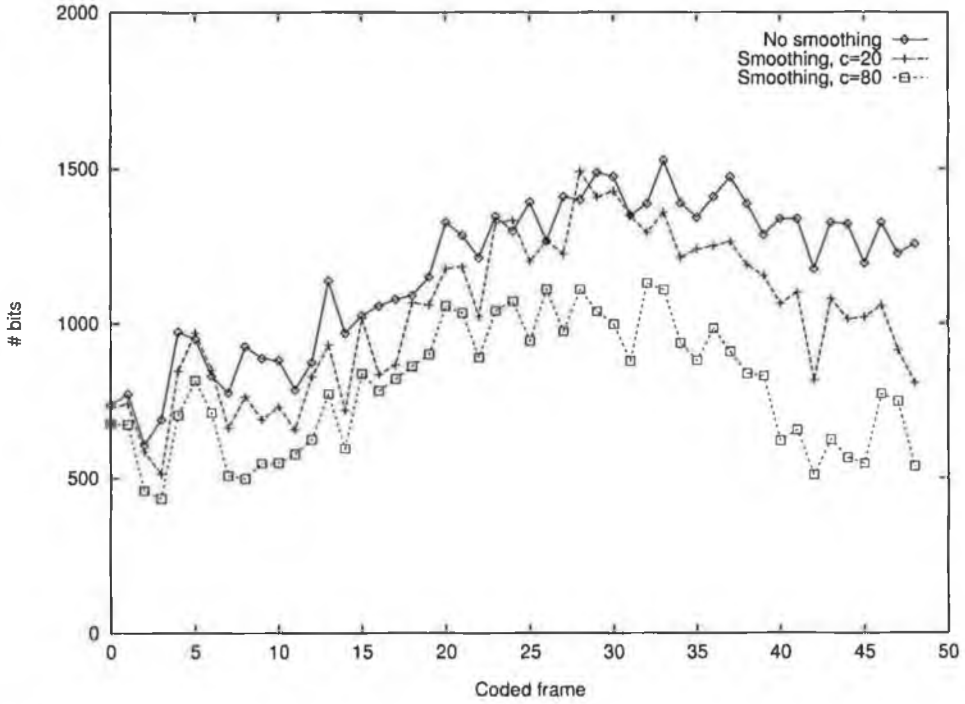


Fig 3.12 : Motion parameter bitcounts (8x8 blocks, constrained model failure area)

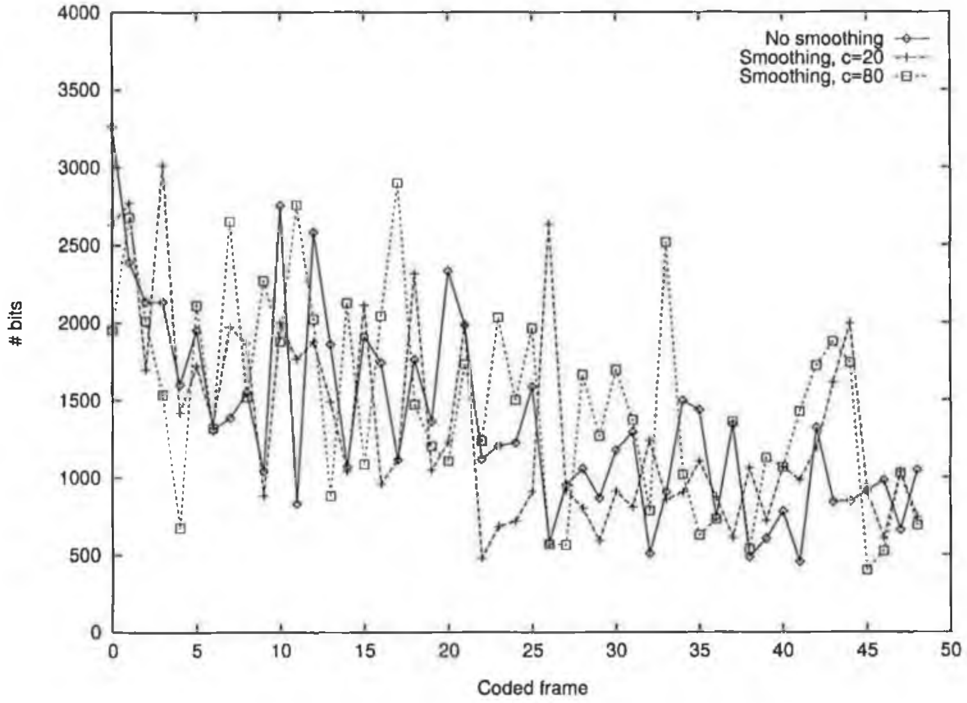


Fig 3.13 : Colour parameter bitcounts (8x8 blocks, constrained model failure area)

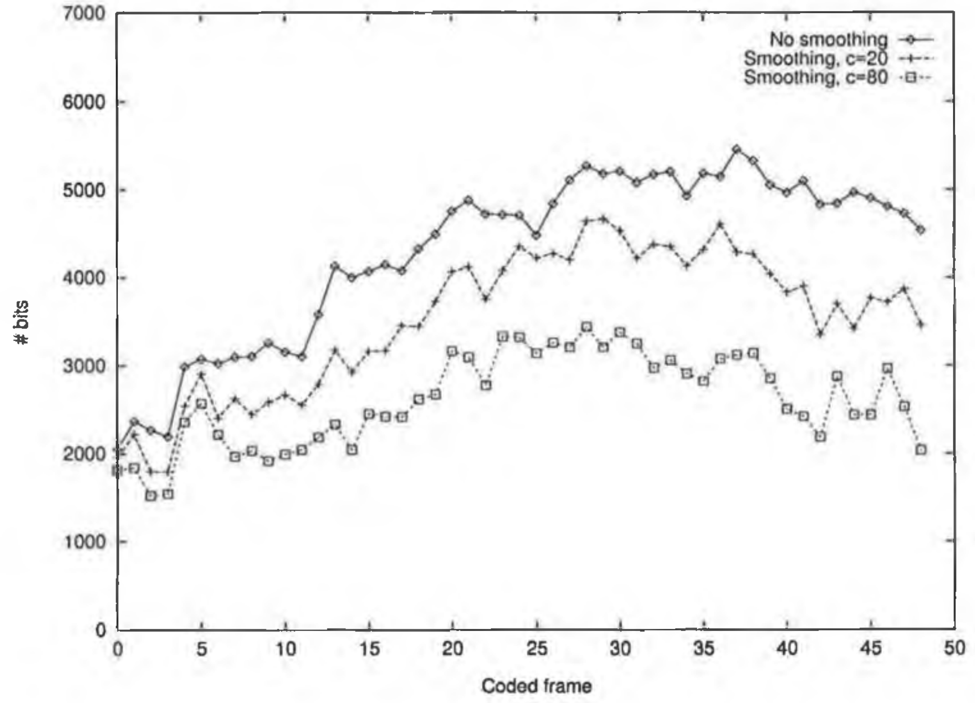


Fig 3.14 : Motion parameter bitcounts (4x4 blocks, constrained model failure area)

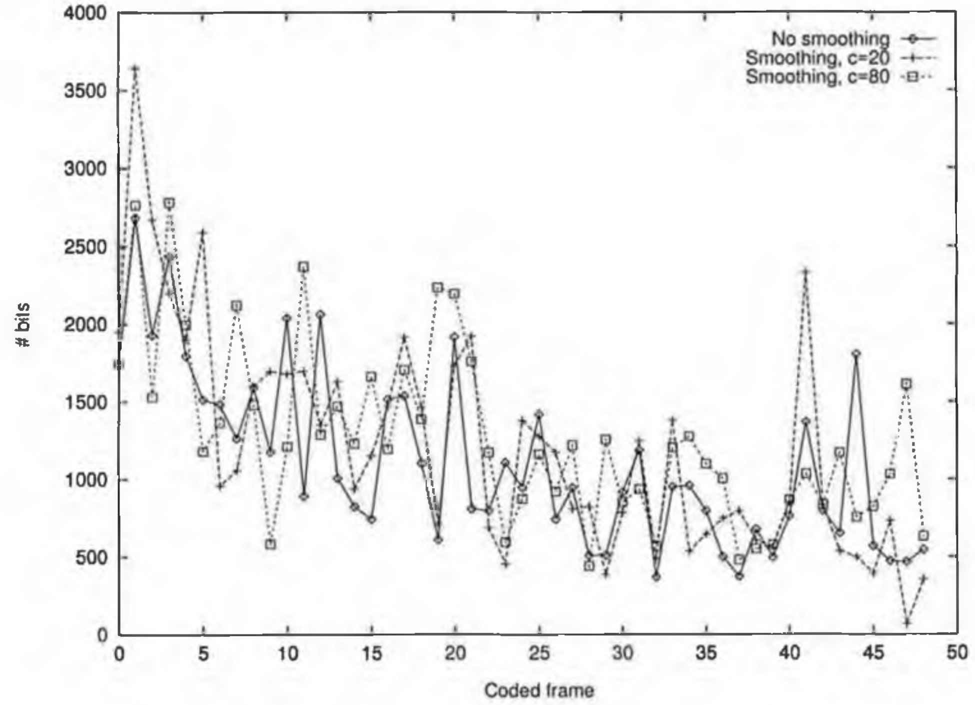


Fig 3.15 : Colour parameter bitcounts (4x4 blocks, constrained model failure area)

The second set of tests give a clearer view of the effect of motion information within SIMOC when model failure detection cannot be relied upon to cater for significant distortion errors. The effects can readily seen in the comparative PSNR measures detailed later in Section 3.2.4

The different MF detection mechanisms used between the two sets of tests also throws up an apparent anomaly in the colour parameter bitcount measures which should be noted. In the original MF detection process, with a defined model compliant variance threshold, less colour parameter bits are required for the initial coded frames than in the constrained MF case. That is because the assumption was made, with the MF pel limit constraint, that more pels than this limit would be classified as MF to maintain overall quality. This is valid in all but the first few frames of the sequence. Tests comparing both the variance threshold and the MF-pel limit were carried out for this test set but effect was found to be minimal.

3.2.4 Comparative SNR measures

The direct influence of grid density on picture quality within SIMOC is illustrated in fig. 3.16. These results are for constrained MF area with no smoothing applied and the increase in bit rate is due solely to coding the extra motion vectors. It can be seen however that on doubling the vector grid density an average increase of 1 to 2 dB gain in PSNR is achievable.

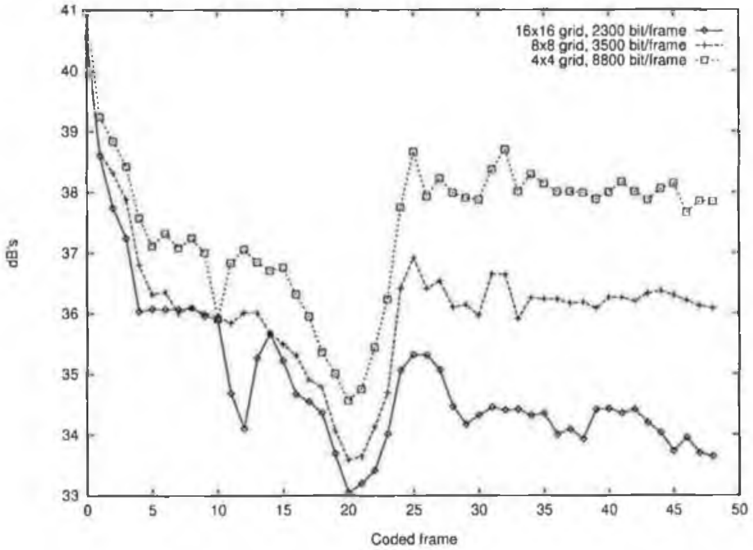


Fig 3.16 : Luminance PSNR comparison of vector grid densities with constrained model failure area

Focussing on the use of smoothing again and the most likely implementation of this within SIMOC fig 3.17 shows what is achievable at the same bitrate using a smooth 8x8 block grid rather than the specified 16x16 one. A small improvement in SNR (½ to 1dB) is evident.

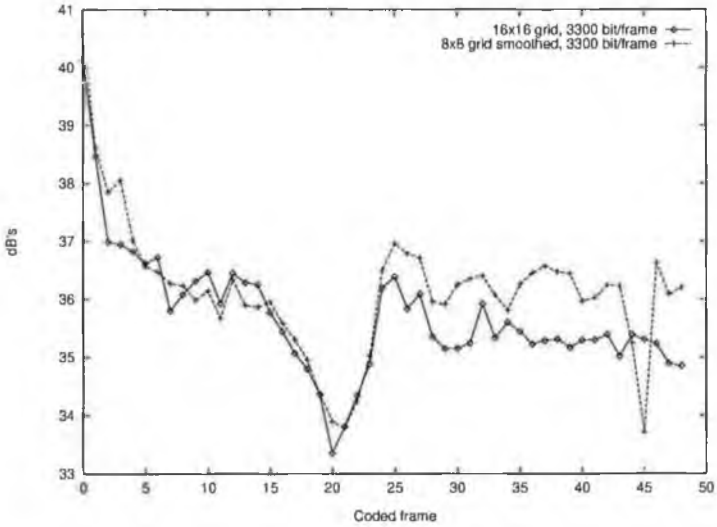


Fig 3.17 : Luminance PSNR comparison of 16x16 grid and smoothed 8x8 grid with same bitrate

If with improved arithmetic coding we can code the denser but smoother grid of motion vectors with the same bitrate as an unmodified 16x16 grid, as is suggested in [23] we could see better improvement. Fig 3.18 illustrates the case.

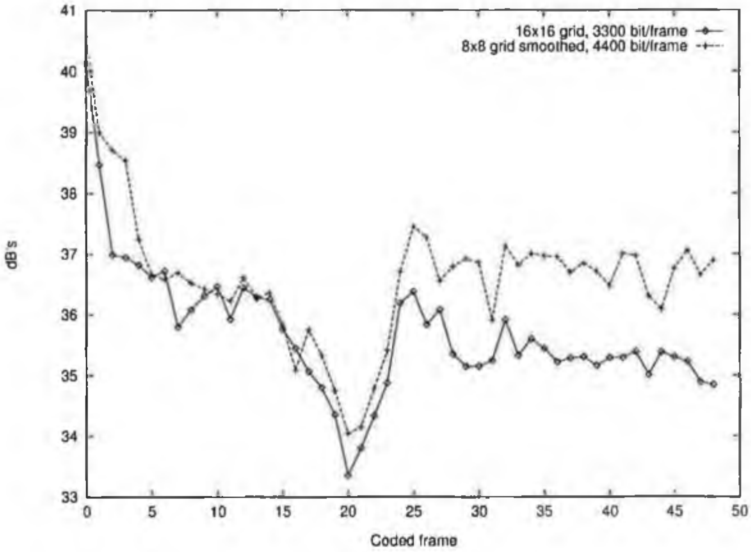


Fig. 3.18 : Luminance PSNR comparison of 16x16 grid and smoothed 8x8 grid with same model failure area.

3.2.5 Conclusions

The source model within SIMOC-1 has proved to be very reliant on good motion estimation for successful and efficient operation. A sparse vector field generated using hierarchical block matching does provide a generally reliable estimate but this can be dramatically improved upon by using fields of higher density. As has been shown in this section, denser fields with better overall motion estimates can be realised without necessarily incurring a large bitrate increase. Marginal improvement in decode picture quality can be shown now at the same bitrate and better predictive arithmetic coding suggests more significant improvements of 1 to 2dB gain in signal to noise ratio could be made at the same overall bitrate.

3.3 Motion estimate accuracy

3.3.1 SIMOC Motion Estimation

Motion estimation within the SIMOC encoder specification uses a 3-stage hierarchical block matching algorithm based on Bierling's work [18]. This involves a progression from a coarse estimation with a large matching window and filtered data to a fine estimate with a smaller window on bi-linearly interpolated data. The last stage yields ½ pel accurate vectors. These had already been shown in MPEG-1 to give improvement over single pel resolution vectors used in earlier algorithms. The stage parameters used are set out below in table 3.7.

Table 3.7 : Stage parameters for hierarchical block matching

Stage	Window size	Step size
1	32x32 window on mean value filtered data	±2 pels first then ±1 pel
2	16x16 window on original unfiltered version	±1 pel
3	16x16 window on bilinearly interpolated versions of original (8x8 window in original sampling grid)	±1 pel (±½ pel step in original)

The search is performed at every 16th pel in every 16th line, within the changed regions as defined by the pre-computed change detection mask, and results in individual motion vectors of ± 4½ pixel range. These vectors are then bi-linearly interpolated to provide a value for the vector field at every pel in the object.

As it stands this approach has a number of problems. The vector range chosen is quite restrictive when it is considered that SIMOC also codes at a reduced frame rate and performs temporal sub-sampling. It is equivalent to a maximum of ±1½ pel per frame of a 25Hz sequence and while this is mostly adequate for the *'Miss America'* test sequence, a more challenging and active sequence such as *'Susie'*, fig. 3.19, poses a major problem. As with most hierarchical schemes, false lock conditions can arise depending on search positions chosen, and as each vector is used in interpolating adjacent vectors an incorrect estimate can adversely affect a large area of the synthesised image. This increases model failures and colour parameter bitcounts.

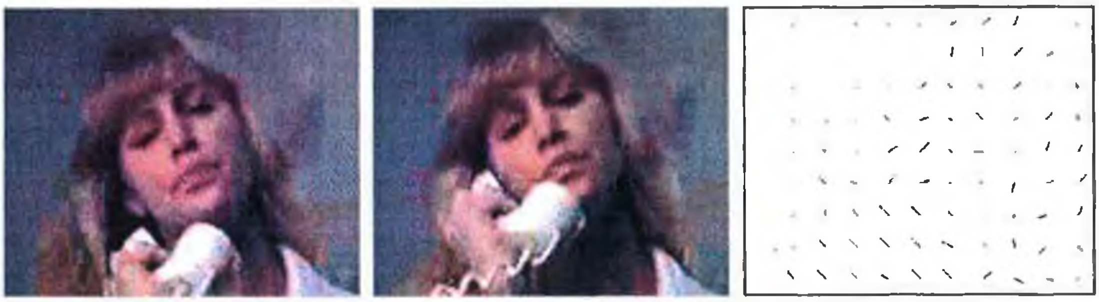


Figure 3.19 : Coded frames from 'Susie' and typical motion vector field for this sequence.

3.3.2 Error Surfaces

The selection criteria for the best block match at each stage of Bierling's scheme is the lowest mean absolute difference (MAD) value. If these MAD values are considered to form an energy surface for each search position, then the aim is to demonstrate a correlation between the characteristics of these energy surfaces and the associated confidence of the chosen best vector in corresponding image regions. In areas of low image detail, such energy surfaces can be shallow and this would indicate the likelihood of false matches using the conventional approach, resulting therefore in an unstable motion vector field. On the other hand, regions with a high degree of textural information will result in energy surfaces with better defined minima, and the standard minimum search should lead to a vector which would have a high degree of confidence associated with it. This research tested these assumptions and results from the *Miss America* test sequence are reported.

3.3.3 New Vectorfield Smoothing Proposal

The smoothing approach proposed here models the motion vector field as a flexible connected grid with stiffness values allocated to the grid branches. The particular stiffness values for any particular branch would be determined by the confidence measure associated with that node point i.e. vector, which in turn would be based on evaluation of local contextual and textural information and adjacent motion vectors. A number of next closest minima positions could also be extracted from the shallow surface regions and these positions tested for associated vector coherence using perhaps a Gibbs / Markov Random Field vector field model similar to that proposed in [23]. In this model, a grid region of high stiffness constraints implies a smoother motion field in the associated image region.

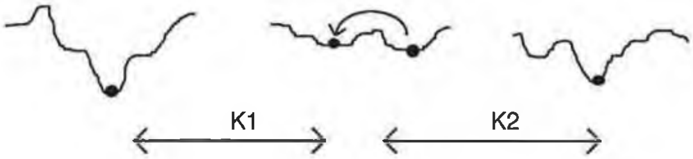


Figure 3.20 - One-dimensional representation of mean absolute difference energy surfaces

Figure 3.20 above illustrates the principle applied on one dimensional data. The first and third samples show deep minima suggesting confident vector matches. The second sample exhibits local minima however and an inappropriate minimum could be chosen depending on the search pattern. Use of the proposed method could result, depending on context, in relatively high stiffness constraints K1 and K2 forcing the vector to a more coherent and accurate position. Due to the expected reduction in vector field entropy, a denser field could also be generated leading hopefully to a closer approximation to the true movement in the sequence.

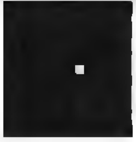
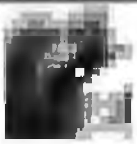
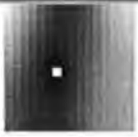
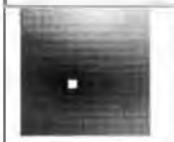
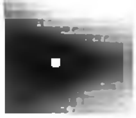
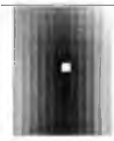
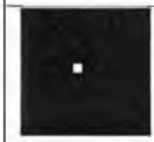
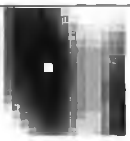
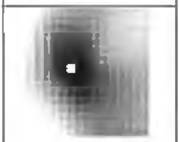
3.3.4 Vector Confidence Measure

The *Hessian* operator was used in examining the correlation between high detail regions and their MAD surfaces. This operator is defined as

$$H(f) = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \qquad \text{Eq 3.5,}$$

and tends towards a maximum in areas of high luminance change in both horizontal and vertical directions. Feature location algorithms make use of this property. Table 3.8 shows results from a ± 8 pel full search on 16x16 blocks of raw image data. The image data in this case is from frames 90 and 93 of Miss America and the Hessian was applied directly to the image data.

Table 3.8 : Low and High Hessian Sample Error Surfaces from 'Miss America'

MAD Surfaces	Histogram Equalized	Average Image Hessian	Maximum Image Hessian	MAD Surfaces	Average Image Hessian	Maximum Image Hessian
		0.3	4		9.9	107.7
		0.2	5.4		18.3	143.4
		0.2	1.9		11.1	212.7
Low Hessian Blocks				**High Hessian Blocks**		

The examples do broadly comply with the original assumptions as to the nature of surfaces, with regions of low average Hessian resulting in shallow energy surfaces and high Hessian regions giving better defined minima. However the surfaces were found to be far more dependent on the nature of the detail in the block rather than the amount of that detail. The second high-Hessian example in table 3.8 has the highest average Hessian in the frame shown and corresponds to the boundary of hair with the left side of Miss America's face. This has a sharp boundary and hence high Hessian but only in one direction and results in a valley shaped minimum region on the MAD surface. This means that the horizontal component of the motion vector is well defined, but the vertical component is not and a false value may easily result. This relationship between image feature orientation and motion accuracy is well established by Kearney et al. [89] and Nagel et al [90].

3.3.5 Surface Types

3.3.5.1 Low Hessian blocks

Figures 3.21 and 3.22 show both types of low Hessian error surfaces encountered. Figure 3.21 has a number of local minima as initially assumed, but figure 3.22 shows a reasonably well defined single minima for an even lower average hession value. Figure 3.23 show a typical image region resulting in a low hessian value.

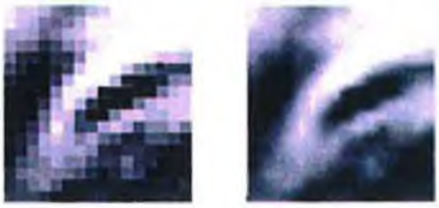


Fig. 3.21 : Block 1 (Avg. Hess = 0.3)

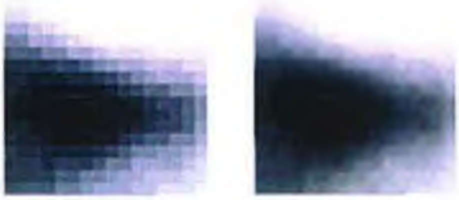


Fig. 3.22 : Block 24 (Avg. Hess = 0.2)

Fig 3.23 : Typical image region resulting in low hessian

3.3.5.2 High Hessian blocks

Two types of high Hessian error surfaces were also observed. Figure 3.24 shows the expected well defined minima but fig 3.25 shows a mimina valley only well defined in one direction. A false y vector component could easily result from such a surface. Fig. 3.26 a) & b) show typical image regions resulting in high hessian values. Fig 3.26 a) would produce a minima valley as seen in fig. 3.25.

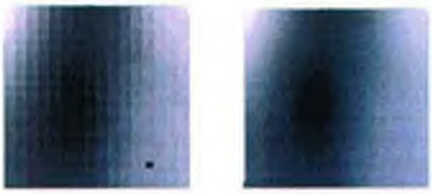


Fig 3.24 : Block 23 (Avg. Hess = 11.1)

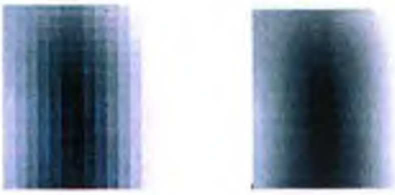


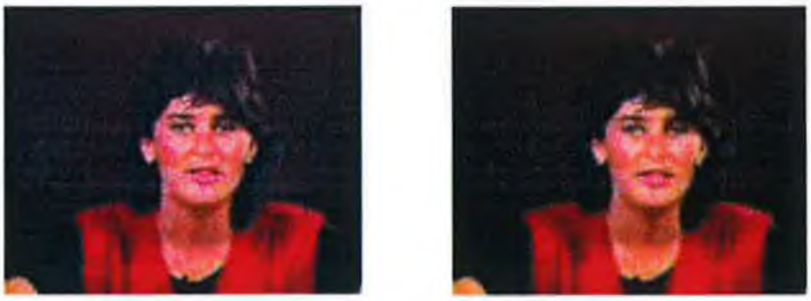
Fig 3.25 : Block 16 (Avg. Hess = 20.6)

a) *b)*

Fig 3.26 : Typical image regions resulting in high hessian

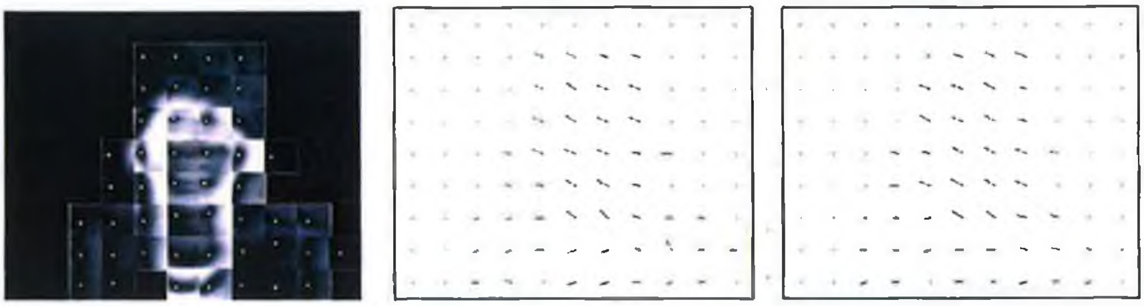
3.3.6 Hessian Smoothing Results

The vector confidence measure was incorporated into the combined DFD / vectorfield model of Stiller and eq. 3.4 by varying the distance, l, between vectors depending on Hessian value of block. l was made inversely proportional to the average Hessian of the block to increase influence of 'confident' vector estimates and decrease influence of 'dubious' ones. Figures 3.27 to 3.28 show results of this approach. The initial estimates to the smoothing algorithm are based on minimum mean absolute difference as calculated for a full ±8 pel search on 16x16 pel blocks, rather than using a hierarchical search. Computational load is therefore much higher for this approach but the overall vector field entropy is reduced after smoothing and spurious vectors such as one within the right shoulder region are seen to be corrected.



a) b)

Fig 3.27 : Miss America a) frame 78 & b) frame 81 of original test sequence.



a) b) c)

Fig 3.28 : a) Associated mean absolute difference surfaces, b) motion vector field and c) field after smoothing

The actual motion in this scene pair is down and to the right. Vectors are reversed on the figures to better indicate true motion for display purposes. Figure 3.28 a) shows the associated MAD surfaces and the minimum points of these surfaces are denoted by a single white pel.

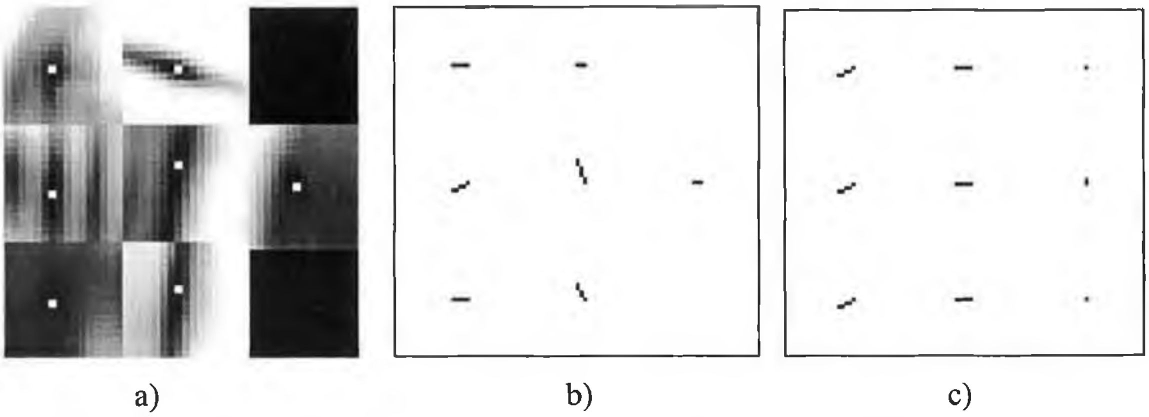


Fig 3.29 : a) Set of MAD surfaces of shoulder region from later frame pair, b) vector fields before and c) after smoothing

Figure 3.29 a) above shows a set of low Hessian value MAD surfaces which have been histogram equalised to better reveal the shallow structure. Figure 3.29 b) shows the initial estimates passed to the smoothing algorithm and figure 3.29 c) the result of smoothing with a high weighting given to the vectorfield model. The central vector is seen clearly to move along the surface minimum valley to a more coherent vector position and illustrates the benefit of this approach.

Figure 3.29 a) also illustrates the existence of local minima which can lead a hierarchical block matching method in the wrong direction at an early stage, and prevent it finding the true lowest minimum. Although it has been shown above that this lowest minimum is not always the best one, it is likely to be a better starting point for an iterative smoothing operation than a secondary minimum.

3.3.7 Conclusions

Two characteristics of block matching MAD error surfaces which can lead to poor vector fields have been identified. One is the existence of local minima which can lead hierarchical search type motion estimators in the wrong direction at an early stage, and prevent location of the true minimum. The second is the common occurrence of valley shaped minimum regions, which leave the motion vector poorly defined along the direction of the valley. The Hessian operator was found to identify deep minima, but further work is needed to recognise and deal correctly with the valley shaped features. Stiller's method has been shown to be an effective means of correcting erroneous vectors in general. It is, however, improved by selecting secondary minima as candidate alternative vector positions, rather than just some function of the neighbouring vectors as he proposed [23], and by modifying the weighting between the displacement and vector field cost functions dependent upon measures of the nature of the MAD error surfaces. My implementation combining vector confidence estimates with Stiller's general smoothing criteria has been successful, if computationally expensive.

3.4 Next Steps

The motion estimation work reported on here was studied within the context of SIMOC but has general applicability to standard block based encoders. Work into optimisation of these and other standard block based techniques for purely compression functionality was also underway in parallel to the object based coding developments. This was to an extent fuelled by the appearance on the market of proprietary codecs working on the public telephone network at lower data bitrates than those targeted by H.261. The next chapter details work carried out by the author as input to the collaborative ITU-T SG15 group to define a near term very low bitrate video coding standard. More generally the aspects of SIMOC relating to content coding - shape coding and image analysis and were further developed within the MPEG-4 and COST 211ter frameworks respectively and the author's work in these areas is described later in the thesis.

4. Enhanced Block-Based Coding

4.1 Introduction

Advances in modem technology in 1995, allowing full duplex data rates of 28.8 and later 33.4 kbit/s on the widely available public switched telephone network (PSTN), increased demand for a new near term low bit rate video coding standard with the goal of improved efficiency below digital network rates. ITU-T fulfilled this demand with the development of Recommendation H.263 building on the well-established, but ISDN-optimised, H.261 algorithm. Despite the advances in object based coding, and the SIMOC-1 developments described in chapter 3, standard block based techniques were still considered at this time, the most generic and generally applicable video compression tools. The macroblock coding structure, intra- and inter-frame block transform coding and displacement vector compensation, were successfully incorporated into a number of existing video coding standards and an optimisation of these techniques was a natural development in the search for higher efficiency.

4.1.1 ITU-T Recommendation H.263

The baseline version of the new standard is quite similar to the existing H.261 standard with the following key differences -

- Motion vectors are produced with half-pixel accuracy. This had already been shown in MPEG-1 to show coding gain.
- Loop filter is not used. The pixel interpolation function involved in the half-pixel motion compensation process has the effect of low pass filtering.
- Motion vector data and overhead information is coded more efficiently.
- Support for more picture formats: from SQCIF (128x96 pixels) through to 16xCIF (1408x1152 pixels).
- Coefficient coding employs a 3-dimensional scheme conveying {run, level and last/not_last} information. H.261 and MPEG-1 used {run, level} coding only.

As well as the differences listed above, the H.263 algorithm also includes a set of more complex and hence optional extensions to improve efficiency even further.

These options, as of May 1995, included:

- *Unrestricted Motion Vectors* (UMV) allowing motion vectors to point outside the frame,
- *Syntax-based Arithmetic Coding* (SAC) giving compression improvement over Huffman coding,
- *Advanced Prediction Mode* (APM) using overlapped block motion compensation and providing for 8x8 block motion vectors and
- *PB-Frames* allowing insertion of MPEG-like bi-directionally interpolated frames to double the framerate at minimal increase of bitrate and delay.

Each optional annex is discussed briefly below.

4.1.1.1 Unrestricted Motion Vectors

Motion vectors are normally restricted to only reference pixels within the picture for use in the compensation process. This option relaxes that restriction and allows vectors to point outside the picture. Where this happens the nearest edge pixel is used in the prediction, found by limiting the x and y vector components to the last pixel position inside the coded area. This option improves prediction at the edges of the picture only when there is significant motion of objects or background into the picture. Motion out of the picture is already handled correctly. Gain from this option is clearly dependant on scene type.

4.1.1.2 Syntax-based Arithmetic Coding (SAC)

Arithmetic coding is a more efficient alternative to the more widely used Huffman variable length coding. SAC is a particular variant of arithmetic coding fitting the syntax of H.263 and directly replaces the existing entropy coder for all coded motion vectors, DCT coefficients, block and macroblock syntax symbols. SAC was investigated by the author and implemented as an ITU-T core experiment during the course of this research and is further discussed in Section 4.2. Software written for this work now forms part of the reference simulation software for H.263.

4.1.1.3 Advanced Prediction Mode

This mode allows the use of up to 4 motion vectors instead of one per macroblock for motion compensation and also defines an *overlapped block motion compensation (OBMC)* technique for predicted images giving a significant subjective gain to output frames. The use of 8x8 block grid motion vectors has already been shown in Chapter 3 to significantly improve motion prediction within an object based coding scheme and the result holds equally well for block based coding. To minimise the increase in bitrate here, the decision is made on a macroblock basis whether to send one vector or four, depending on efficiency gain. The OBMC technique works by building each pixel in an 8x8 prediction block from a weighted sum of neighbouring block prediction values. The motion vectors from the current block and the two nearest blocks are used to form this prediction according to the mask shown in fig. 4.1. The correct weightings for each of the neighbouring block vectors are determined by 'folding over' the associated cross 'limbs' of fig. 4.1. E.g. the pixel at position (1,1) of the current block, with (0,0) representing top-left, has a weighting of 5 for the prediction using the current block vector, a weighting of 1 for the prediction using the vector above and 2 for the prediction using the vector to the left. The weighted prediction results must then be normalised to normal pixel range by dividing by 8.

				1	1	1	1	1	1	1	1				
				1	1	1	1	1	1	1	1				
				1	1	2	2	2	2	1	1				
				2	2	2	2	2	2	2	2				
1	1	1	2	4	5	5	5	5	5	5	4	2	1	1	1
1	1	2	2	5	5	5	5	5	5	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	5	5	5	5	5	5	2	2	1	1
1	1	1	2	4	5	5	5	5	5	5	4	2	1	1	1
				2	2	2	2	2	2	2	2				
				1	1	2	2	2	2	1	1				
				1	1	1	1	1	1	1	1				
				1	1	1	1	1	1	1	1				

Figure 4.1 : Weighting mask for overlapped block motion compensation

4.1.1.4 PB-Frames

The name of this new picture type derives from the P and B frames of MPEG, already described in Chapter 2. The coding overhead of sending multiple P and B frames is reduced in this option by treating a P and B frame as one unit as shown in fig. 4.2.

The flexibility of MPEG is lost but the result is an effective doubling of framerate with only a modest increase in bitrate.

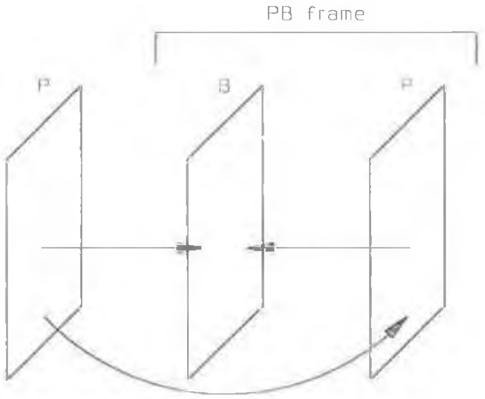


Figure 4.2 : Prediction of PB-frames

4.1.1.5 Optional annex performance

Table 4.1 below illustrates the benefits, in relation to coding efficiency, of each option applied individually to three different types of video scene. Individual option contributions are scene dependant and some benefit coding efficiency directly and others indirectly by improving subjective performance. UMV, for example, works best in scenes with significant motion at the edges of the picture involving movement of objects or background into the picture, such as with the *Foreman* sequence involving large pans and camera shake, but for all scene types this option generally gives an efficiency improvement. PB-frames on the other hand generally increase bitrate by a modest amount but by doubling the framerate give a significant subjective improvement. They are not, however, effective for sequences with very fast or complex motion or low initial frame rates.

Table 4.1 : Coding efficiency gains of H.263 (May '95) optional annexes relative to baseline H.263 (All sequences ~28.8 kbit/s average, 12.5 fps QCIF)

Sequence (type)	Quantiser	UMV	SAC	APM	PB-frames
Claire (static, central motion)	6	+0.7%	+1.7%	+9.1%	-7.4%
Mother & Daughter (static, more complex motion)	9	+1.4%	+3.9%	+5.5%	-9.0%
Foreman (very fast, complex motion)	20	+9.4%	+6.1%	-1.9%	-25.7%

The performance of H.263 is dramatic compared to H.261 and fig. 4.3 gives an indication of improved picture quality at the same bitrate. Table 4.2 and fig. 4.4 also gives objective performance measure figures for H.263 with all options enabled, H.263 base level and H.261 at ~9.6, ~14.4 and ~28.8 kbit/s; the nominal PSTN modem data rates. Results show that H.263 achieves a higher SNR than H.261 for all sequences and bit-rates tested. In addition, there is considerable subjective improvement.

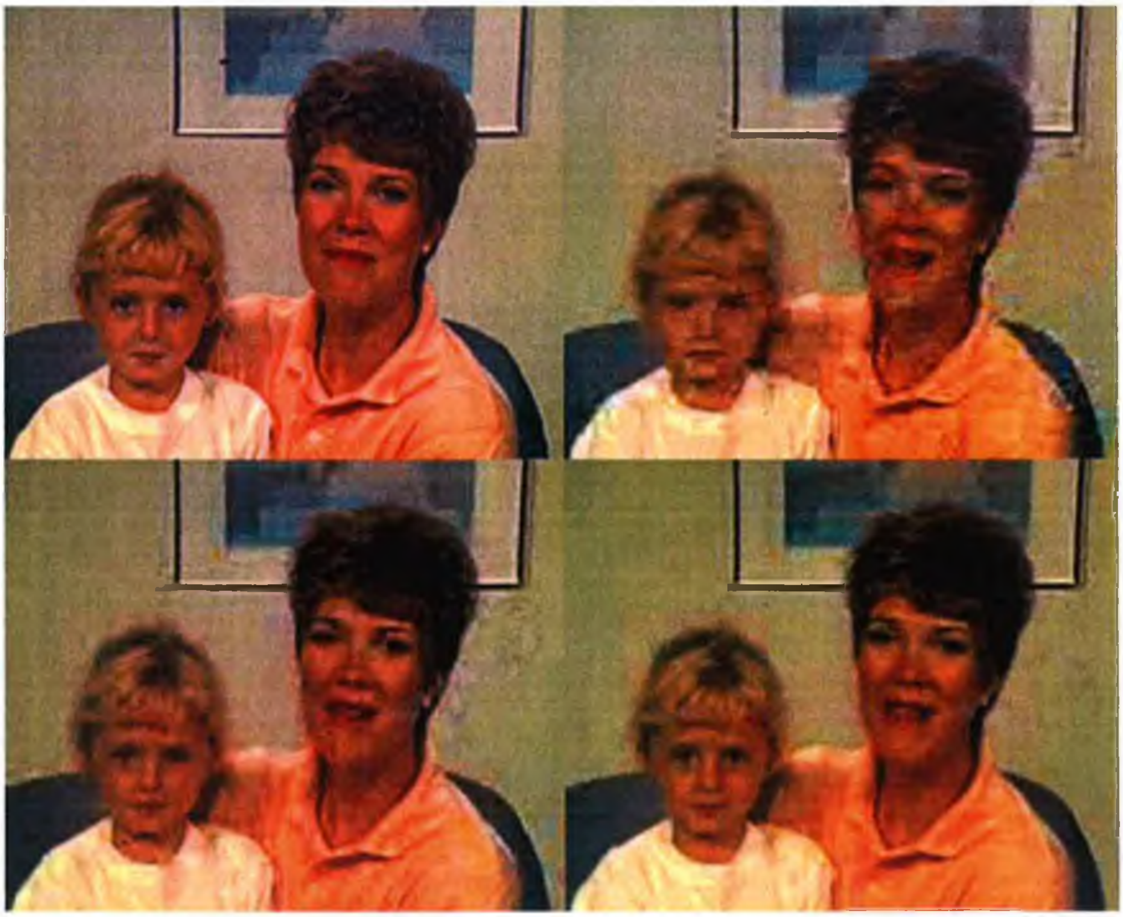


Figure 4.3 : (Clockwise from top-left) Still from Mother & Daughter Original, Coded H.261, Coded H.263 (all annexes on) and Coded H.263 baseline. All coders targetting bitrate of 28.8 kbit/s

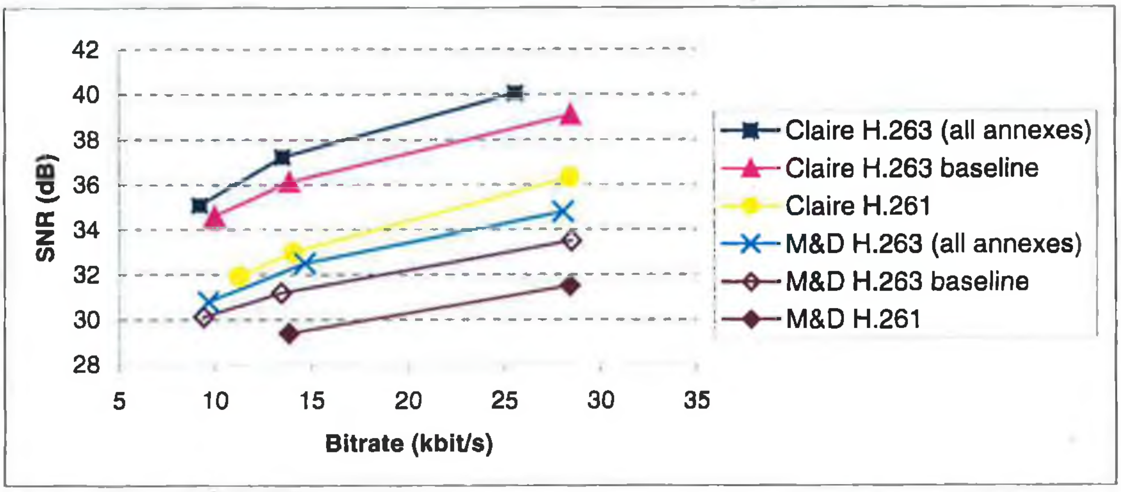


Figure 4.4 : PSTN Rate-Distortion Comparison between H.261 and H.263 Claire and Mother & Daughter sequences

Table 4.2 : Performance of H.261 against baseline H.263 and also H.263 with all optional annexes turned on.

	H.261			H.263 Baseline			H.263 With all options		
Sequence Name	Q' (Avg.)	Y SNR (dB)	Bitrate (kbit/s)	Q	Y SNR (dB)	Bitrate (kbit/s)	Q	Y SNR (dB)	Bitrate (kbit/s)
Claire			#	13	34.62	9.97	12	35.08	9.20
	21	33.01	14.05	10	36.12	13.84	8	37.23	13.49
	11	36.32	28.41	6	39.11	28.46	5	40.06	25.57
Mother &			#	18	30.12	9.40	16	30.79	9.67
Daughter	26	29.40	13.83	14	31.20	13.42	11	32.51	14.67
	14	31.51	28.41	9	33.51	28.49	7	34.78	28.07
Susie			#	20	29.78	9.61	23	29.54	9.67*
	28	29.33	15.83	19	30.04	14.63	22	29.77	14.52*
	23	30.11	29.18	14	31.51	29.27	13	31.62	28.47
Foreman			#	29	26.43	9.32	31	26.65	9.98*
	25	28.08	15.67	18	28.58	14.57	18	28.97	14.37*
	29	27.49	30.67	20	28.22	27.79	18	28.54	27.57

* PB-frames not used for these sequences as the frame rate is too low.
# Bit-rate target could not be met by H.261

Test conditions used in table 4.2 generation slightly favour H.263. It was not possible to turn off H.261 buffer regulation and quantiser values quoted are averages over the entire sequence. Fairer conditions at ITU-T meetings have shown an average increase of 2 to 3 dB SNR of H.263 over H.261 at the same bitrate and in terms of efficiency gain it is generally agreed that H.263 achieves approximately the same subjective quality of H.261 at half the bitrate.

4.1.2 H.263+

After May 1995, and in collaboration with the ISO/IEC MPEG-4 standard development, a number of further optional extensions to H.263 were proposed and adopted by ITU-T in the so called H.263+ [24] project. These, from [24], include -

- *Advanced INTRA Coding* improving the compression efficiency for INTRA macroblock encoding by using spatial prediction of DCT coefficient values;
- *Deblocking Filter* which reduces the amount of block artifacts in the final image by filtering across block boundaries using an adaptive filter;
- *Slice Structured Mode* which allows a functional grouping of a number of macroblocks in the picture, enabling improved error resilience, improved transport over packet networks, and reduced delay;
- *Reference Picture Selection* which improves error resilience by allowing a temporally previous reference picture to be selected which is not the most recent encoded picture that can be syntactically referenced;
- *Reference Picture Resampling* which allows a resampling of a temporally previous reference picture prior to its use as a reference for encoding, enabling global motion compensation, predictive dynamic resolution conversion, predictive picture area alteration and registration, and special-effect warping;
- *Reduced-Resolution Update.* A mode which allows an encoder to maintain a high frame rate during heavy motion by encoding a low-resolution update to a higher-resolution picture while maintaining high resolution in stationary areas;
- *Independent Segment Decoding* which enhances error resilience by ensuring that corrupted data from some region of the picture cannot cause propagation of error into other regions;
- *Alternative INTER VLC* which reduces the number of bits needed for encoding predictively-coded blocks when there are many large coefficients the block; and

- *Modified Quantization* which improves the control of the bit rate by changing the method for controlling the quantizer step size on a macroblock basis. This reduces the prevalence of chrominance artifacts by reducing the step size for chrominance quantization, increases the range of representable coefficient values for use with small quantizer step sizes, and increases error detection performance and reduces decoding complexity by prohibiting certain unreasonable coefficient representations.

Other enhancements include improved PB-frames, custom source formats and provision of supplemental picture information. Three new picture types are also supported in H.263+, specifically for spatial and temporal scalability, including -

- *B-frames* : Pictures having two reference pictures, one of which temporally precedes the B picture and one of which is temporally subsequent to the B picture;
- *EI-frames* : Picture having a temporally simultaneous reference picture; and
- *EP-frames* : Pictures having two reference pictures, one of which temporally precedes the EP picture and one of which is temporally simultaneous.

Two of the most effective of the efficiency related options, Syntax-based Arithmetic Coding and Advanced Intra Prediction were implemented and further developed by the author during the course of this compression efficiency research work and are described in detail here. Contributions in these areas included software to the ITU-T SG XV body and submissions to the WG11 MPEG standards group.

4.2 Syntax-based Arithmetic Coding

4.2.1 Arithmetic Coding Background

When using Huffman coding an integral number of bits must be used in codewords representing the message symbols. This means that Huffman coding performs optimally only in the cases that symbol probabilities are integral powers of ½. This is not normally the case and Huffman coding can take up to 1 extra bit per symbol. Table 4.3 illustrates an extreme example of divergence from entropy when using Huffman coding. From eq. 2.1 Entropy = 9000*0.152 + 500*4.323 + 500*4.323 =

5691 bits. Huffman coding however would yield 9000*1 + 500*2 + 500*2 = 10000 bits.

Table 4.3 : Sample message source with Huffman codes

Symbol	No. Occurrences	P_i	$\log_2(1/P_i)$	Huffman Code
a	9000	0.9	0.152	0
b	500	0.05	4.323	10
c	500	0.05	4.323	11

Arithmetic coding combines individual message symbols into a single data unit to be coded whilst still allowing coding and decoding on a symbol by symbol basis. The average number of bits for a given symbol over a long stream of data can then be non-integer and can approach the actual entropy measure. It works by considering each symbol as a half-open subinterval in the interval [0,1). (E.g. a = [0,0.2), b = [0.2,0.3) etc.). Subsequent symbol combinations form subintervals of the previous subinterval (e.g. aa = [0,0.04), ab = [0.04,0.06), ba = [0.2,0.22) etc.). Fig. 4.5 illustrates this better with the message and symbol probabilities used in the Huffman coding example of Chap. 2. Both encoder and decoder know probability distributions a priori. Hence, by assigning enough precision bits to a real number representing the subinterval that particular subinterval can be determined from any other and the corresponding message symbols can be decoded. Compression and symbol by symbol coding is achieved by sending the smallest number of bits needed to uniquely identify each subinterval and hence the message to that point.

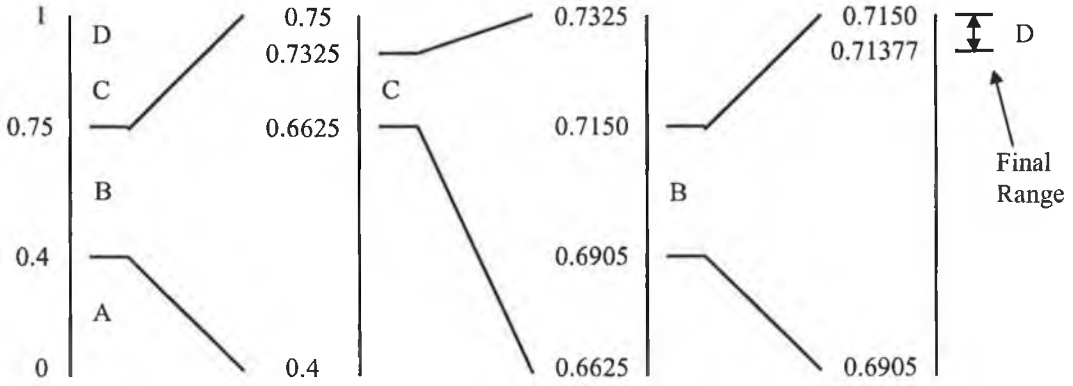


Figure 4.5 : Representation of arithmetic coding process for message {B,C,B,D} with interval scaled up at each stage. D acting as end of message marker.

Like Huffman coding, the more probable symbols correspond to larger subintervals and thus require fewer bits of precision to identify the [low, high) range. It is also not necessary to send both ends of the range to the decoder. However, a single number may be ambiguous; e.g. 0.0 could represent a, aa, aaa … in the example above. Hence, the decoder does require a special end-of-message symbol known to both encoder and decoder.

Software implementations of arithmetic coding must also address the problems of incremental transmission/reception and limits in real number precision as symbol probabilities multiply indefinitely. Both are addressed by a re-normalisation process whenever the range gets too small. Multiplication by two is a common software technique equating to a bit shift of the low and high values. Zeros are shifted into the low order bits of *low* and ones are shifted into *high*. As the code range narrows the top bits of low and high become the same and will not be changed by future narrowing of the range. They can now be transmitted immediately.

4.2.2 Syntax-based Arithmetic Coding for H.263

Syntax-based Arithmetic coding follows the syntax of the H.263 Huffman VLC tables for the Picture, Group of Blocks and Macroblock layers. The block layer as shown in Fig. 4.6 is extended to include $TCOEF_1$, $TCOEF_2$, $TCOEF_3$ and $TCOEF_r$ which are the possible 1st, 2nd, 3rd and rest of the Last-Run-Level DCT coefficient symbols respectively. In fig. 4.6 the extra transform coefficient symbols are only present when one, two, three or more coefficients are present in the block layer, respectively.

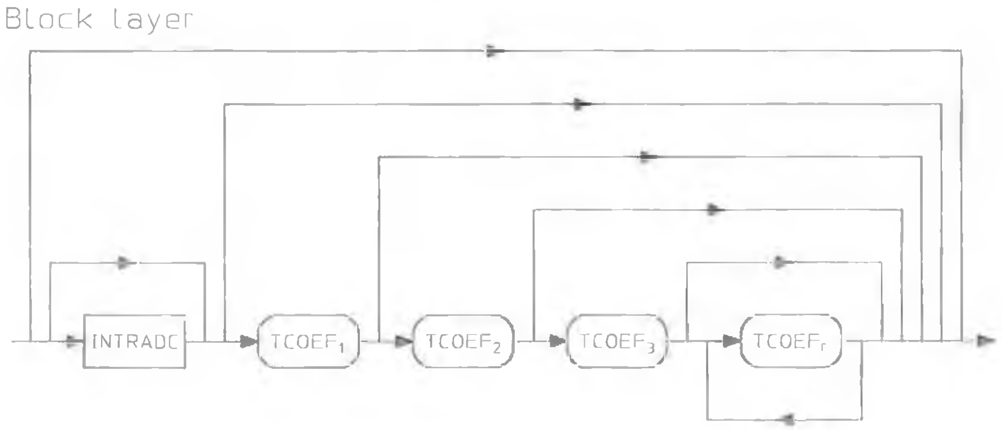


Figure 4.6 : Structure of SAC Block layer

Predefined static models for these and all the other syntax elements are provided by H.263 and known by both encoder and decoder. These were calculated for both intra and inter coding symbols produced during coder runs with the ITU-T test video material. Code for SAC was written by the author and incorporated into the H.263 reference software and table 4.4 shows typical results compared to H.263 Huffman coding. Gains over the standard VLC coding are highest for the entirely intra coded frames but overall gains are still significant.

Table 4.4 : SAC coding results for 30 frame CIF sequences, fixed Q=10

Sequence	**1st Intra**	**Avg. Inter**	**1st Intra (Gain)**	**Avg. Inter (Gain)**	**Overall Gain**
Mother & Daughter	47288	1767	41936 (+11%)	1762 (+0.3%)	+5.6%
Coastguard	89152	23831	84616 (+5.1%)	23088 (+3.1%)	+3.3%
Stefan	111168	36567	98440 (+11.4%)	35545 (+2.8%)	+3.6%
	Huffman Coding bits		**SAC Arithmetic Coding bits**		

4.2.3 Adaptive SAC

One of the advantages of arithmetic coding is the ability to change the probability model of a symbol set independently of the coding process. Adaptive models at both encoder and decoder can update probability distributions based on the symbol frequencies seen so far in the message and can achieve substantial extra compression over use of static fixed models as used in H.263 SAC. An adaptive form of SAC was implemented and the results in table 4.5 obtained.

Table 4.5 : SAC vs Adaptive SAC coding tests on full length QCIF sequences

Sequence	Huffman	SAC	Adaptive SAC
Mother & Daughter	120775 (960)	115623 (+4.3%)	112043 (+7.2%)
Foreman	141048 (399)	134434 (+4.7%)	131850 (+6.5%)
	Bytes (No. frames)	**Bytes (Gain)**	**Bytes (Gain)**

The gain over SAC is considerable, particularly over long video sequence runs. These gains also compare favourably, c.f. [81], with results obtained using the binary arithmetic Q-coder [25] on the MPEG-4 test material. The binary Q-coder is also used in ISO/JPEG [26] and ISO/JBIG [82].

Some problems were evident, however. The encode/decode process of adaptive SAC is slower due to recalculation of model probabilities. To increase speed probability distribution updates can be calculated after N new events rather than after every one. The probabilities are now not so closely tracked so some inefficiency will be introduced. A second more important problem with the adaptive approach is that probability tables of both encoder and decoder must be kept synchronised otherwise decoder will output completely erroneous data. Any transmission error would be fatal to adaptive SAC streams. As H.263 was intended for use over narrow bandwidth and potentially error prone network links this last observation was significant and is the reason the adaptive form of SAC was not used. Possible workarounds for this would have involved signalling in the associated control protocols (e.g. H.245 [28]) to revert to fixed tables in the presence of errors but the extra complexity / gain argument was not won in this case. However it should prove useful in applications where transmission errors can be corrected and where adequate processing power is available.

4.3 Advanced Intra Prediction

In the baseline H.263 recommendation bits spent on intra macroblocks account for a high proportion of the total bitrate for a typical video coding session. Collectively they make up all of the coded I-frames and a variable but often significant portion of P and PB-frames. Each intra macroblock is coded independently, but experience in MPEG-1 and JPEG had already shown that previous block average (DC term) prediction was advantageous in general to intra block coding efficiency. The block spatially preceding the block to be coded was found often to be a good predictor.

A number of intra block prediction strategies were further investigated by the author during the course of both ITU-T H.263 development and that of ISO/IEC MPEG-4 video, discussed in chapter 5. These included spatial prediction and DCT coefficient prediction and the results reported below are drawn from work done while participating in core experiments in both these bodies.

4.3.1 Spatial Prediction

Real world imagery contains many examples of large scale flat zones or common background texture which can be exploited even when performing an intra coding update. DC level prediction from the previous block is used in MPEG-1 and JPEG intra coding and the idea behind spatial prediction was to extend this average level only prediction to whole block texture prediction in the spatial domain. The first intra prediction technique tested was in fact a combination of two techniques first proposed by Faramarz Azadegan of GTE and Gisle Bjontegaard of Telenor. The GTE proposal [29] used previously decoded macroblocks above and to the left of the macroblock to be coded as predictions and coded in normal inter mode. The predictor which minimised the resulting bitrate was chosen and signalled at macroblock layer. This method performs well in cases of large horizontal or vertical structures. The Telenor proposal [30] worked at the block level (8x8) fitting a plane to the previously decoded edge pixels of the blocks above and to the left. This method represents a good prediction of the DC and first two AC coefficients in the zig-zag scan and generally performed better than the GTE method. The technique actually implemented was a combination of these two proposals and is illustrated in fig. 4.7. Here the GTE method was modified to use predictors P3 and P4 rather than P1 and P2 and even one stage further to use just the 8 closest pixel values of each predictor H(i) and V(j).

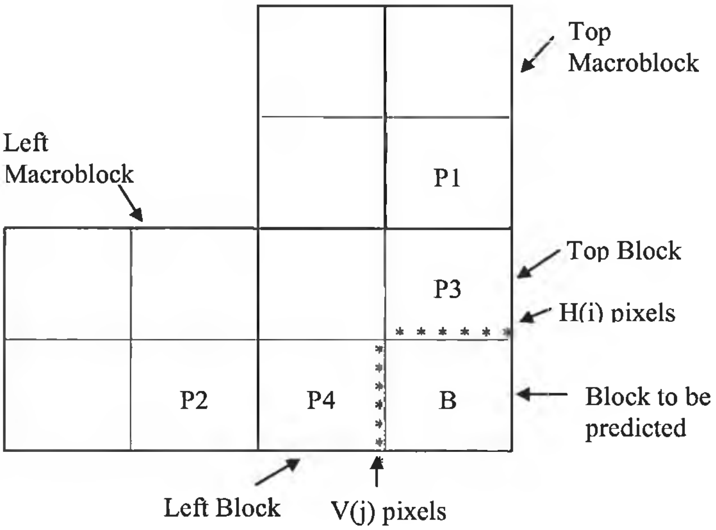


Figure 4.7 : Candidate predictors for use in Block B coding

The Telenor method was simplified by removing the slope calculation and predicting just the DC coefficient. Elimination of the slope calculation reduced efficiency slightly (~2%) but reduced complexity was judged to be more beneficial. All possible predictors then reduced to:

- $\mathrm{Pred}_1(i,j) \;=\; h(i) \quad i,j = 0..7$ from P3
- $\mathrm{Pred}_2(i,j) \;=\; v(j) \quad i,j = 0..7$ from P4
- $\mathrm{Pred}_3(i,j) \;=\; \frac{1}{16}\sum_{k=0}^{7}\big(h(k)+v(k)\big) \quad i,j = 0..7$ from P3 and P4

if only one of h(k) or v(k) is inside the picture Pred_3 would reduce to :

- $\mathrm{Pred}_3(i,j) \;=\; \frac{1}{8}\sum_{k=0}^{7}h(k) \;\text{ or }\; \frac{1}{8}\sum_{k=0}^{7}v(k) \quad i,j = 0..7$ from P3 or P4.

A fourth option would be the standard intra coding mode.

Table 4.6 shows some initial results applied to intra frames. First frames of the MPEG-4 test sequence set were chosen in these tests.

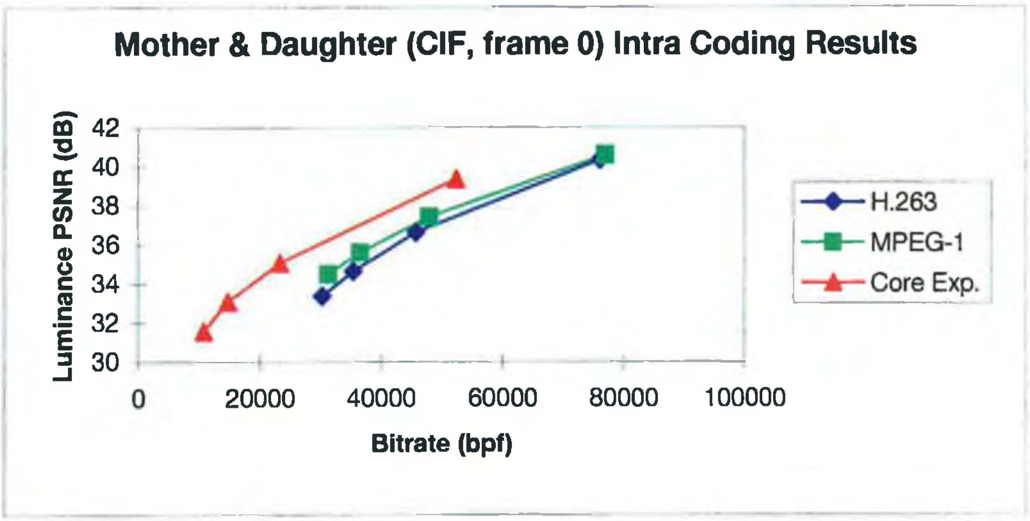
Table 4.6 : SNR and bitrate reductions relative to baseline H.263. Two quantiser results shown.

I-frame Source	SNR Drop	Bitrate Reduction	SNR Drop	Bitrate Reduction
Container Ship	0.01 dB	29 %	0.44 dB	45 %
Akiyo	0.33 dB	35 %	0.68 dB	54 %
Mother & Daughter	0.32 dB	42 %	0.81 dB	63 %
Hall Monitor	0.34 dB	35 %	0.65 dB	49 %
	Q = 10, CIF		**Q = 20, CIF**	

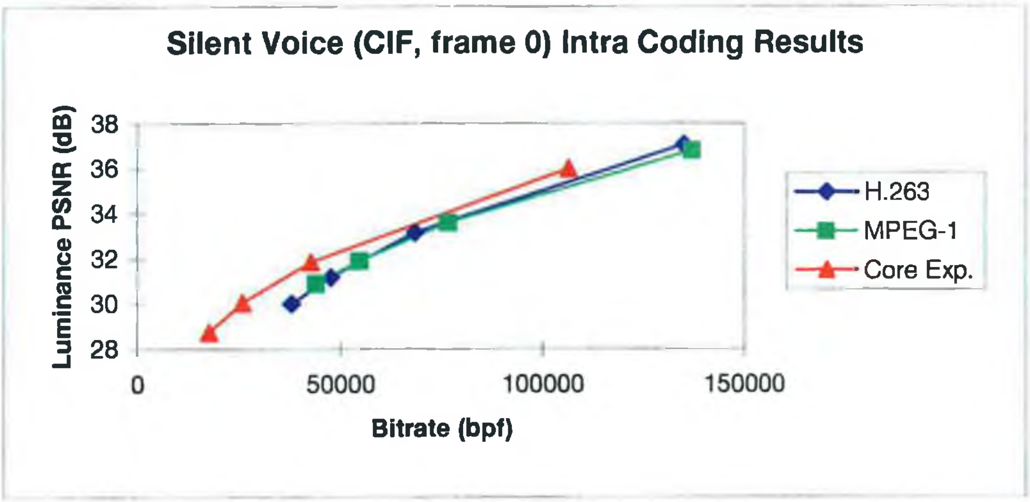
A submission by the author to MPEG-4, [31] and Appendix A, details further tests, using more sequences at QCIF and CIF resolutions and showing rate distortion curves. Fig. 4.8 shows the main results here with rate distortion comparisons between H.263, MPEG-1 (which uses DC term prediction) and the spatial prediction technique described above. The experiment was implemented within the latest version, at that time, of the Telenor H.263 simulation code (tmn v. 1.7).

The new intra coding scheme could choose four different coding modes for each macroblock and two extra bits of syntax per macroblock were needed to signal which mode was chosen. These extra bits were taken into account in the results shown.

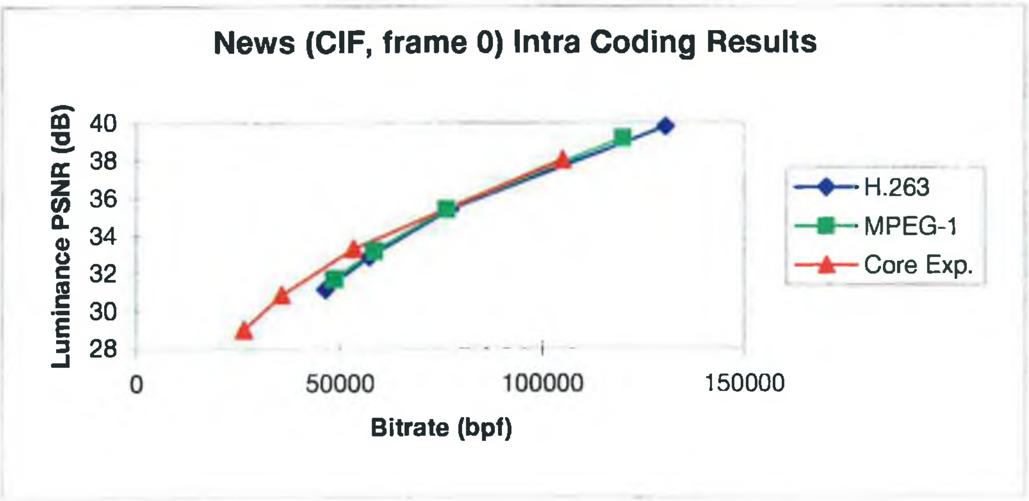
Results were obtained for the CIF first frames of the following test sequences - *Mother and Daughter, Hall monitor, Coastguard, Silent voice* and *News*. The luminance PSNR and bitcounts were plotted for quantisation parameter values of 5, 10, 15 and 20.


Mother & Daughter (CIF, frame 0) Intra Coding Results
Luminance PSNR (dB)
42
40
38
36
34
32
30
0
20000
40000
60000
80000
100000
Bitrate (bpf)
H.263
MPEG-1
Core Exp.

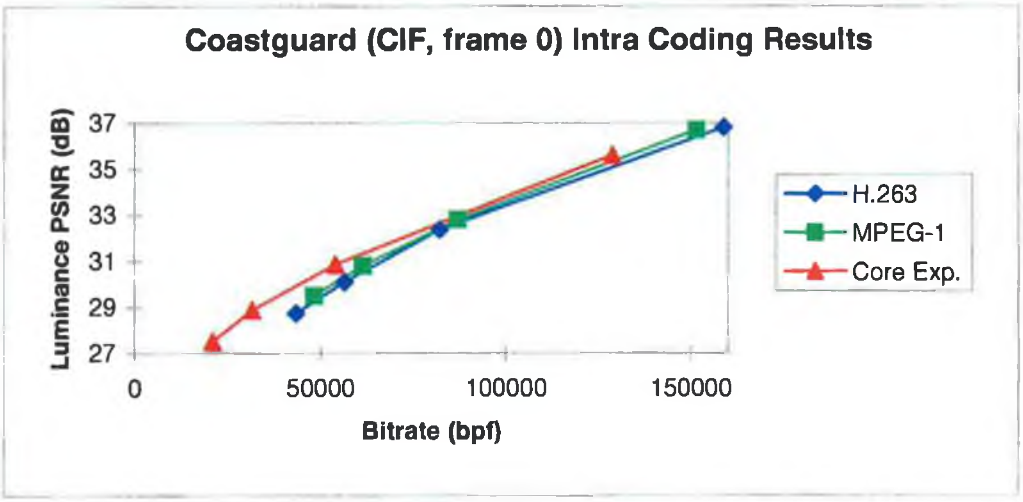
a)


Silent Voice (CIF, frame 0) Intra Coding Results
Luminance PSNR (dB)
38
36
34
32
30
28
0
50000
100000
150000
Bitrate (bpf)
H.263
MPEG-1
Core Exp.

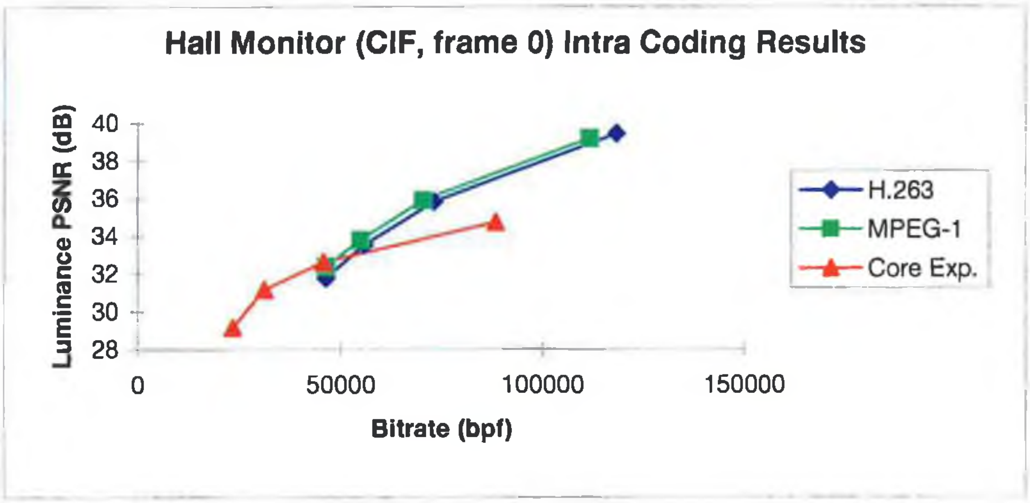
b)


News (CIF, frame 0) Intra Coding Results
Luminance PSNR (dB)
40
38
36
34
32
30
28
0
50000
100000
150000
Bitrate (bpf)
H.263
MPEG-1
Core Exp.

c)



d)



e)

Figure 4.8 a - e : Intra coding rate distortion curves for CIF first frames.

Table 4.6 shows quite impressive bitrate reductions accompanied by small SNR drops and despite one anomalous result in fig. 4.8 c) the rate-distortion curves show clear SNR gains at the same bitrate even over MPEG-1 intra prediction. Reduction is greatest at higher quantiser values. Despite these good results, it was decided to investigate further to see if prediction in the DCT domain would perform even better and the next section details these results.

4.3.2 DCT Domain Prediction

This particular method was proposed by T.K. Tan and S.M. Shen of Matsushita in [32]. Instead of a single DC prediction mode as in MPEG-1, one of three modes is used for each macroblock. Figure 4.9 shows three 8x8 blocks of quantised DCT levels labeled A(u,v), B(u,v) and C(u,v), where u and v are the indices for the horizontal rows and vertical columns, respectively.

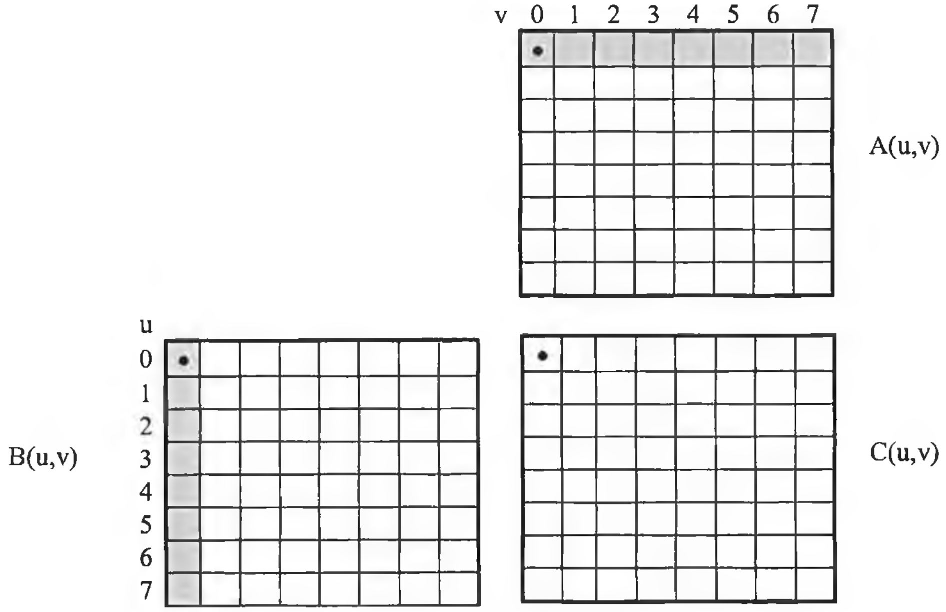


Figure 4.9 : Three Neighbouring Blocks in the DCT domain.

Let C(u,v) be the block to be coded and $E_i(u,v)$ the prediction error, for mode i, formed by subtracting the prediction from blocks A(u,v) and/or B(u,v). The prediction modes are as follows:

- mode 0: DC prediction only.

  $E_0(0,0) = C(0,0) - (A(0,0) + B(0,0))//2$,and
  $E_0(u,v) = C(u,v)$ $u \neq 0, v \neq 0, u = 0..7, v = 0..7$

- mode 1: DC/AC prediction from the block above.

  $E_1(0,v) = C(0,v) - A(0,v)$ $v = 0..7$, and
  $E_1(u,v) = C(u,v)$ $u = 1..7, v = 0..7$

- mode 2: DC/AC prediction from the block to the left.

  $E_2(u,0) = C(u,0) - B(u,0)$ $u = 0..7$, and
  $E_2(u,v) = C(u,v)$ $u = 0..7, v = 1..7$

At the boundary of the frame, if the block above or block to the left of the block being coded does not fall into the frame then the value of 128 and 0 are substituted for the DC and AC component of these blocks, respectively.

The mode selection is done by evaluating the absolute sum of the prediction error, $\mathrm{SAD}_{\mathrm{mode}\ i}$, for the four luminance blocks in the macroblock and selecting the mode with the minimum value.

$$\mathrm{SAD}_{\mathrm{mode}\ i} = \sum_b \left[E_i(0,0) + \sum_u |E_i(u,0)| + \sum_v |E_i(0,v)| \right], \quad i = 0..2,\ b = 0..3,\ u,v = 1..7$$

The value of the mode is transmitted in the macroblock data according to the following syntax in Figure 4.10. The mode is coded using variable length code in table 4.7. One mode is transmitted per macroblock.

DC/AC_MODE	MCBPC	CBPY	DQUANT	Block Data

Figure 4.10 : Structure of macroblock layer in I-frames

Table 4.7 : VLC table for DC/AC_MODE (for I-frames)

Index	DC/AC Prediction Mode	VLC Code
0	0	0
1	1	10
2	2	11

One particular result set drawn from [33] is reproduced here in table 4.8 and fig. 4.11 for comparison with the spatial prediction method. Here, VM stands for the MPEG-4 verification model which defaulted to H.263 for this mode. The DC only prediction represents MPEG-1 and JPEG prediction. The spatial prediction data was derived independently following the description in section 4.3.1 and the DC/AC prediction was the new proposal from [32].

Table 4.8 : Complete result set for Mother and Daughter (CIF, frame 0).

	MPEG-4 VM		DC only Prediction			Spatial Prediction			DC/AC Prediction		
QP	bits	PSNR	bits	PSNR	Δbits	bits	PSNR	Δbits	bits	PSNR	Δbits
5	78142	40.72	70976	40.65	-9.2%	52642	39.61	-32.6%	61333	40.65	-21.5%
6	67419	39.70	59801	39.59	-11.3%	41100	38.14	-39.0%	51248	39.59	-24.0%
7	59464	38.70	51402	38.59	-13.6%	35055	37.52	-41.0%	43437	38.59	-27.0%
8	54091	38.01	45768	37.91	-15.4%	29184	36.56	-46.0%	38045	37.91	-29.7%
9	49935	37.33	41353	37.18	-17.2%	25725	36.07	-48.5%	33849	37.18	-32.2%
10	46267	36.79	37353	36.63	-19.3%	22085	35.34	-52.3%	30363	36.63	-34.4%
11	43449	36.29	34264	36.11	-21.1%	19834	34.97	-54.4%	27739	36.11	-36.2%
12	41018	35.85	31663	35.68	-22.8%	17641	34.41	-57.0%	25637	35.68	-37.5%
13	39208	35.40	29698	35.22	-24.3%	16127	34.16	-58.9%	23712	35.22	-39.5%
14	37205	35.07	27563	34.84	-25.9%	14348	33.64	-61.4%	22038	34.84	-40.8%
15	35660	34.73	25894	34.48	-27.4%	13371	33.41	-62.5%	20706	34.48	-41.9%
16	34135	34.47	24264	34.23	-28.9%	12026	32.98	-64.8%	19500	34.23	-42.9%
17	33043	34.15	23074	33.88	-30.2%	11300	32.74	-65.8%	18512	33.88	-44.0%
18	32003	33.92	21941	33.64	-31.4%	10524	32.45	-67.1%	17661	33.64	-44.8%
19	31020	33.65	20821	33.36	-32.9%	9964	32.26	-67.9%	16888	33.36	-45.6%
20	30385	33.43	20098	33.15	-33.9%	9318	31.99	-69.3%	16226	33.15	-46.6%

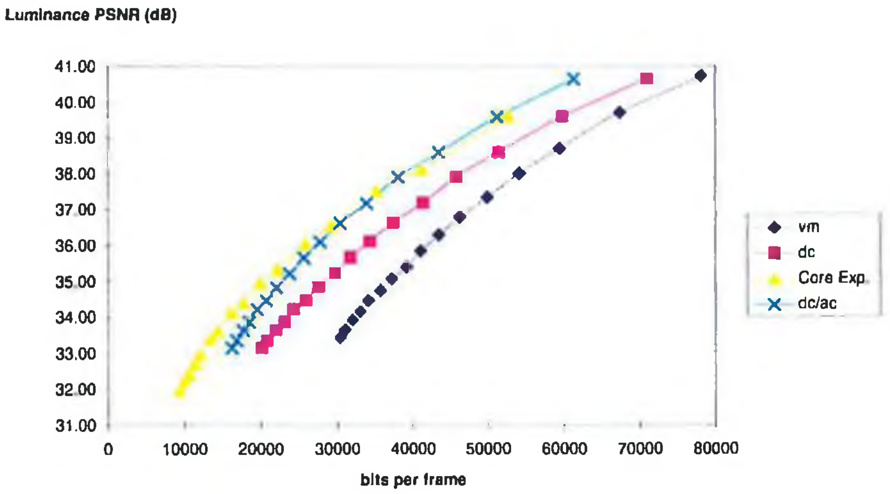


Figure 4.11 : Rate distortion curve for intra prediction tests on Mother and Daughter, CIF frame 0.

These results are typical of the rest of the result set contained in [33] and validate the spatial prediction tests performed earlier. Both DCT and spatial domain prediction outperform standard H.263 and MPEG-1 intra prediction modes both objectively and subjectively for frames generated with the same bitcount. Spatial prediction generally performs well and is better in a rate distortion sense than DC/AC prediction particularly at the higher quantiser values. In objective terms, however, the spatial prediction results do introduce visual artefacts. A key difference between spatial and

DC/AC prediction is that the latter uses prediction after quantisation and therefore does not introduce any loss in SNR or picture quality. Spatial prediction before quantisation can result in vertical or horizontal streaking artefacts as residual errors are quantised away and not coded.

DCT domain prediction forms the basis of Annex I of H.263 and is also incorporated into MPEG-4 video. Several modifications and further improvements were also made including modified inverse quantisation and separate VLC for intra coefficients, two extra coefficient scans in addition to the zig-zag scan and a process of *oddification* applied to DCT coefficients, i.e. addition of 1 to any even valued coefficient, in order to minimise the impact of IDCT mismatch errors. Inverse quantisation of the INTRADC coefficient is modified to allow a varying quantisation step size, unlike in the main part of H.263 where a fixed step size of 8 is used for INTRADC coefficients. Inverse quantisation of all INTRA coefficients is performed without a "dead-zone" in the quantiser reconstruction spacing. The additional scans are also shown below in fig. 4.12. If the vertically adjacent block is chosen for the prediction, the inference is that the block to be coded will be dominated by stronger horizontal frequency content. Scanning pattern A then scans the stronger horizontal frequencies first. Similarly, for the case where the horizontally adjacent block is chosen, scanning pattern B scans the stronger vertical frequencies prior to the horizontal ones.

1	2	3	4	11	12	13	14
5	6	9	10	18	17	16	15
7	8	20	19	27	28	29	30
21	22	25	26	31	32	33	34
23	24	35	36	43	44	45	46
37	38	41	42	47	48	49	50
39	40	51	52	57	58	59	60
53	54	55	56	61	62	63	64

(a) Scanning pattern A - (Horizontal first)

1	5	7	21	23	37	39	53
2	6	8	22	24	38	40	54
3	9	20	25	35	41	51	55
4	10	19	26	36	42	52	56
11	18	27	31	43	47	57	61
12	17	28	32	44	48	58	62
13	16	29	33	45	49	59	63
14	15	30	34	46	50	60	64

(b) Scanning pattern B - (Vertical first)

Figure 4.12 : Alternate DCT Scanning patterns for Advanced INTRA Coding

4.4 Conclusions

The block based and entropy coding techniques described here and in Chapter 2 represent over two decades of standards algorithm development. The exploitation of image data statistics and human visual system characteristics with these techniques has proved successful. There has, however, also now been general agreement in the video coding community that we are approaching a limit in terms of further coding efficiency gain through optimisation of these techniques. The model and object based techniques, described in Chapters 2 and 3, were thought to hold the best promise of further gains in this area, representing, as they did, a deeper understanding of images as projections of the real world. SIMOC exploited this higher abstraction of images in terms of further compression but it was also thought that this shift away from purely block based coding to content coding would allow many more applications that would need to manipulate that video content. This became the driver for the emerging MPEG-4 multimedia coding standard, described in the next chapter, and the focus for the rest of the research into content-based coding.

5. Content coding with MPEG-4

5.1 Background

MPEG-4 [34, 35, 36] is an ISO/IEC multimedia coding standard now close to final definition by the Moving Picture Experts Group (MPEG). This committee represents over 200 companies directly or indirectly involved in the digital audio-visual industry and has already produced two very successful digital video and audio coding standards: MPEG-1 and MPEG-2. These have addressed compression for storage and retrieval of video and audio on compact disc at up to 1.5Mbit/s and higher quality digital broadcast applications at 4Mbit/s and above respectively. An MPEG-3 project was launched with the focus of high definition television but this requirement was later incorporated into an MPEG-2 profile. Indeed both MPEG-1 and MPEG-2 have been used in many more scenarios than those initially targeted including digital video camcorders and on the internet as a widely used file format. The new MPEG-4 standard has evolved with many more interactive forms of audio-visual entertainment such as games, video-enabled internet, and education in mind and, in addition to higher compression, the standard is unique in supporting new content-based tools for communication, access and manipulation of digital audio-visual data. Video content-based coding techniques are a key enabler of these new capabilities.

Initially the MPEG-4 standard was intended to be a purely higher compression version of MPEG-1 and MPEG-2 with better video quality at lower bitrates, and the MPEG-4 project title remains “very low bitrate audio-visual coding”. It was not proven, however, that significant improvement in compression performance over existing standards was possible and at the same time the demands of digital media users were changing with the phenomenal growth of the internet and advances in generally available computing power. The MPEG-4 group modified its targets considerably and identified eight key functionalities which were not thought to be well supported by existing or other emerging standards and considered useful to a wide range of future audio-visual systems. These were grouped into the three fundamental categories of *Content-Based Interactivity, Compression* and *Universal Access* listed in table 5.1. Several of these functionalities were also addressed in other internet and telecommunication bodies and the group pursued a close alignment to other major standards projects such as H.263 [15] and VRML [38].

Table 5.1 : MPEG-4 coding functionalities

Categories	Functionalities
Content-based Interactivity	Content-based multimedia data access tools Content-based manipulation and bit stream editing Hybrid natural and synthetic data coding Improved temporal random access
Compression	Improved coding efficiency Coding of multiple concurrent data streams
Universal access	Robustness in error-prone environments Content-based scalability

The content-based functionality set is the most innovative aspect of the new standard. It allows end users higher levels than ever of interaction with the actual audio-visual content with applications ranging from user re-composition of virtual scenes through to selection of key audio-visual objects for preferential treatment in terms of bitrate or error robustness. Many of these possibilities are enabled by the new content-based coding approach to compressing video which codes each object separately in such a way that they can subsequently be extracted from the compressed bitstream *without* having to decompress it all first. The advantage is that the separation of objects that may have been used in the production process through techniques such as chroma-keying (blue-screen) is retained even up to the point of display, so even the end user can still interact with the objects.

Efficient compression of video material is still an important goal and MPEG-4 incorporates the best techniques of video compression from MPEG-2 and H.263 but optimising over a much wider bitrate range than its predecessors. Arbitrarily shaped video and composite graphics and real video material (also natural and synthetic sound) are also now handled efficiently. Rather than generate and combine graphics in the studio and then compress the resulting complex image, graphic descriptors are sent directly to the end receiver which then renders the graphic, and composes with other objects such as moving video. Benefits are higher compression, and ability to tailor the graphics to the end user e.g. different subtitle language versions.

Section 5.2 gives an overview of the standard and its component specifications focussing on the visual representation aspects. The new content-based techniques incorporated in the video coding technical specification are further detailed in the following section 5.3 and include shape coding and shape adaptive DCT. Many of the frame based coding efficiency extensions worked on as part of this research within MPEG-4 are common to H.263 and H.263+ and have already been described in chapter 4.

5.2 Standard Overview

The full MPEG-4 standard is comprised of six parts. *Systems*, *Audio* and *Visual* form the core technical decoding specifications. Other parts include a *Conformance Testing* specification, a technical report on *Reference Software* implementations and a *Delivery Multimedia Integration Framework* definition (DMIF). The Systems, Audio, Visual and DMIF specifications will form the core of MPEG-4 Version 1 and are currently at *Final Draft International Standard* (FDIS) status. These will not now change dramatically before full *International Standard* (IS) status by early 1999 however there will also be a backward compatible MPEG-4 Version 2 to follow one year later.

5.2.1 Systems, Audio and Visual

Audio [34] and Visual [35] will define a standardised coded representation of audio and visual content, both natural and synthetic, called "audio-visual objects" or AVOs. Systems [36], will standardise the composition of these objects together to form compound AVOs (e.g. an audio-visual scene), and multiplex and synchronise the data associated with individual objects, so that they can be transported over networks at appropriate quality of service levels. Fig 5.1 below shows the typical view of an MPEG-4 AVO systems receiver.

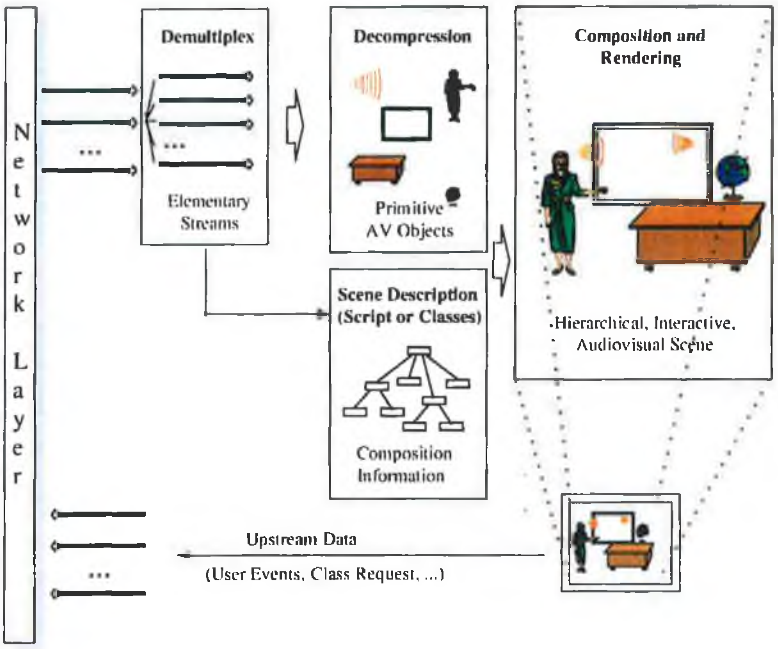

Network Layer
Demultiplex
Elementary Streams
Decompression
Primitive AV Objects
Scene Description (Script or Classes)
Composition Information
Composition and Rendering
Hierarchical, Interactive, Audiovisual Scene
Upstream Data
(User Events, Class Request, ...)

Fig 5.1 : Systems view of an MPEG-4 Receiver. (from [37])

The scene description part of the Systems specification will describe exact spatio-temporal positioning information for all coded AVOs. The format of the description was based on the work of the Virtual Reality Modelling Language (VRML) group although MPEG-4 extends VRML functionality significantly. Java, a recent addition to the Systems layer with the MPEG-J specification, will also be used in the scene interaction model.

5.2.2 Natural Video Coding

Natural video coding in MPEG-4 is supported at a range of bitrates and associated capabilities. Fig. 5.2 below illustrates this and shows a very low bitrate video (VLBV) core providing algorithms optimised for video bitrates in the range of 5 to 64 kbit/s, maximum of CIF resolution video sequences and low frame rates. Support for higher bitrates and broadcast interlaced formats is provided by extensions to this core and bitrates up to 4 Mbit/s are envisioned. Similarly support for content based functionalities and scalability is provided by more complex extensions.

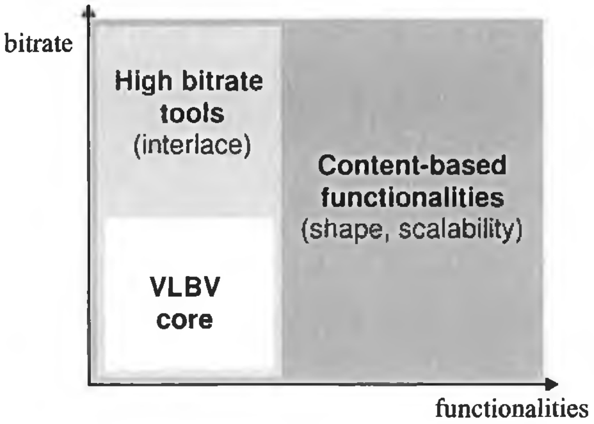

bitrate
High bitrate tools (interlace)
VLBV core
Content-based functionalities (shape, scalability)
functionalities

Fig 5.2 : Natural video coding tools and algorithms in MPEG-4 video

Currently there are two visual Verification Model (VM) specifications; the video VM for natural video coding, frame based or arbitrarily shaped; and the Synthetic/Natural Hybrid Coding (SNHC) VM for synthetic 2D/3D graphics tools. As the tools have been tested and proven within these VMs, they have been progressively incorporated into the single visual draft specification containing all the coding tools relating to visual data.

For the highest efficiency in natural video compression, the video VM [44] built on the work of ITU-T Recommendation H.263, described in chapter 4, and the core visual standard ensures bitstream compatibility with baseline H.263 in frame based operation. Within the generic coder structure, the addition of arbitrary shape and transparency information of so called *Video Object Planes*, see fig. 5.3, enables the range of new content-based functionalities outlined in table 5.1. Shape coding is performed using a bitmap based technique known as context-based arithmetic coding, detailed in section 5.3.1.

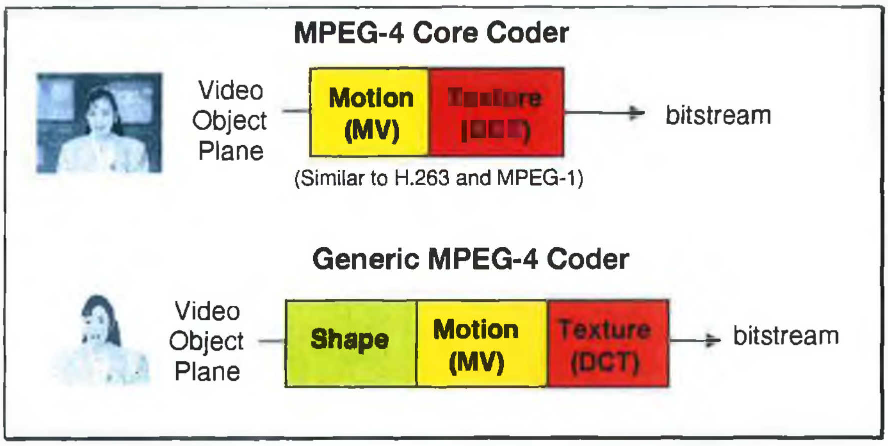


Figure 5.3 : Video Verification Model (VM) stages

5.2.3 Synthetic / Natural Hybrid Coding (SNHC)

The synthetic video components of the SNHC VM currently include media integration of text and graphics (MITG), face and body animation, texture coding (generic and view-dependent textures) and static and dynamic mesh coding with texture mapping. *MITG* provides the capability to overlay and scroll text, images and graphics on coded video backgrounds. *Face Animation* will allow definition and animation of synthetic 'talking heads' as shown in figures 5.4 and 5.5. Currently this is supported through the definition of 68 face feature points which can be animated. Only the face definition parameters (FDPs) and the face animation parameters (FAPs) need standardising here. Work at BT Laboratories [39, 40], combining face model animation with texture mapping, has shown how realistic these synthetic personae can actually be. *Body Animation* issues will also be similarly supported this time with body definition and animation parameters (BDPs and BAPs).

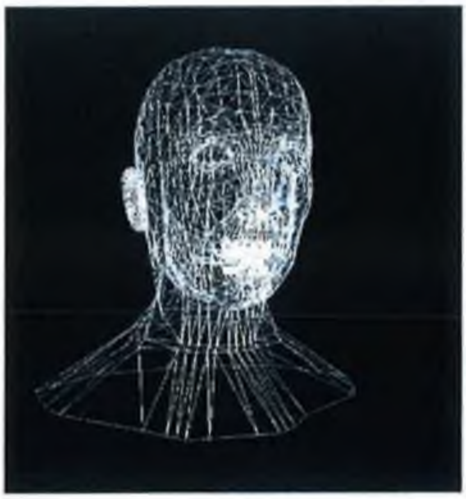


Figure 5.4 : Wireframe head model

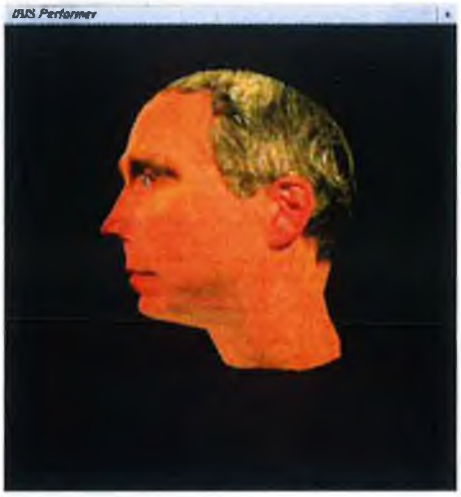


Figure 5.5 : Texture mapped model

Mesh coding will also be supported within the synthetic video toolbox and an example of a static 2D mesh is shown in fig. 5.6. General 3D model representations are also expected to be supported.

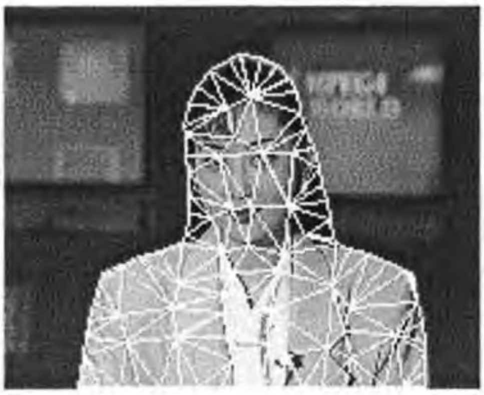


Figure 5.6 : 2D mesh modelling of Akiyo Video Object

5.3 Content Coding Techniques

5.3.1 Shape Coding

There are two types of shape information that MPEG-4 supports; binary and greyscale and both are described in the standard in terms of alpha planes. Binary information denotes object shape and location whereas greyscale information adds support for transparency of objects for composition and blending capability. Binary alpha masks are coded using a modification of the Group 4 facsimile transmission technique known as Context-based Arithmetic Encoding (CAE) and described in T.82/JBIG [41] and in MPEG contributions [42, 43]. Greyscale alpha masks on the other hand are coded using motion compensated DCT in a similar manner to texture coding. An overview of both is given in the following sections. Detailed descriptions are given in [46].

5.3.1.1 Binary Mask Coding

For binary mask coding the following stages are carried out :

- Alpha plane formation and partition into 16x16 Binary Alpha Blocks (BABs)
- BAB coding mode decision
- BAB coding for Intra or Inter-VOPs by CAE

Alpha plane and BAB formation involves extending the bounding rectangle surrounding the VOP shape to multiples of 16x16 blocks. The extension is done from the bottom right corner. All extended alpha samples are set to zero and processing is continued on a BAB by BAB basis.

Each BAB is further subdivided into sixteen 4x4 pixel blocks which are used in the BAB coding mode decision process. Within an intra VOP the three allowed BAB modes are all_0, all_255 and *intraCAE*. For an inter VOP four addition modes cater for various combinations of motion compensated shape coding with no update required and coding with *interCAE*. IntraCAE mode uses only pixels within the current block. InterCAE mode also uses pixels from the previous BAB.

Context-based Arithmetic Encoding (CAE) is used to code each binary pixel of the BAB. For each pixel of an intra coded BAB a 10-bit context $C = \sum c_k \cdot 2^k$ is built as illustrated in fig. 5.7 (a). A 9-bit context is built for inter coded BABs using the current BAB and the motion compensated version of the corresponding BAB in the previous frame as shown in fig 5.7 (b).

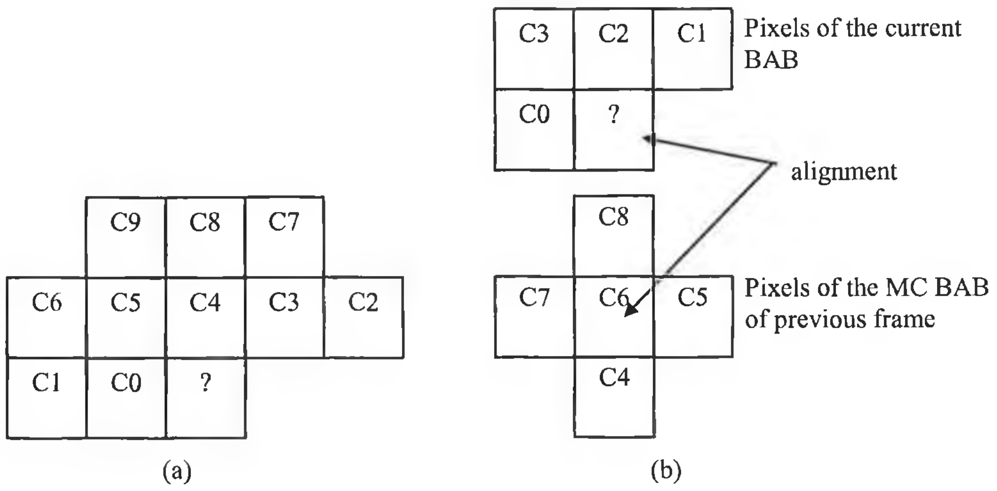

C9
C8
C7
C6
C5
C4
C3
C2
C1
C0
?
(a)
C3
C2
C1
Pixels of the current BAB
C0
?
alignment
C8
C7
C6
C5
Pixels of the MC BAB of previous frame
C4
(b)

Figure 5.7 : (a) The intra context template. (b) The inter context template. The pixel to be coded is marked with '?'

For the intra case, C7, C3 and C2 may not be available at decode time so values are padded from left to right i.e. if C7 is unknown when context is being constructed C7 is set to C8, if C3 is unknown C3 is set to C4 and if C2 is unknown C2 is set to C3.

Similarly for the inter case, if C1 is unknown C1 is set to C2. All other positions are known or can be assumed to be zero.

Given these contexts a probability distribution can be determined for a large sample set of shapes and appropriate probability distribution tables constructed for a multi-symbol arithmetic encoder. As with H.263 and SAC a start code emulation avoidance mechanism is also used in the arithmetic encoder.

5.3.1.2 Greyscale Alpha Map Coding

For greyscale alpha map coding the following steps are taken.

- Alpha plane formation and partition into 16x16 Binary Alpha Blocks (BABs) as with binary coding case.
- Partition of Grey-level alpha map into support function (binary equivalent) and texture. See fig. 5.8
- Coding of support function BABs as in binary mask coding
- Coding of texture as in normal luminance coding.

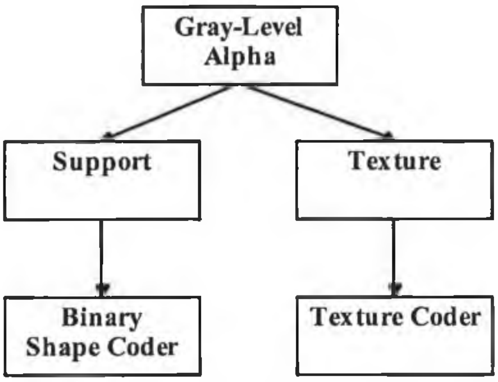
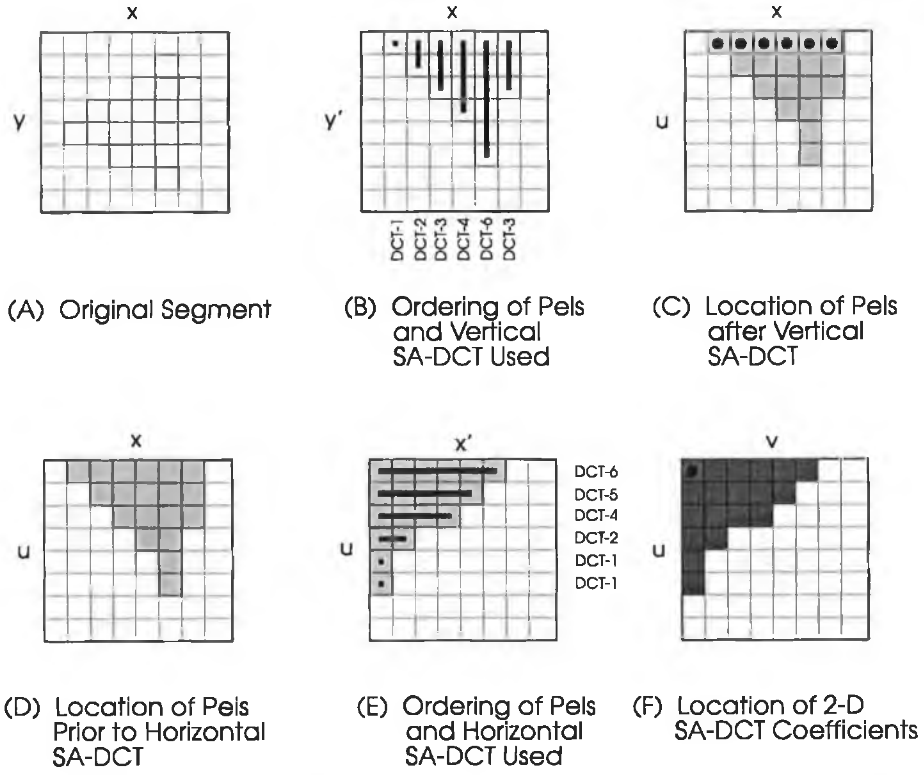


Figure 5.8 : Grey level alpha plane coding process

The grey level alpha coding process is then a combination of binary coding for the support map and texture coding for the alpha values. Some grey level alpha maps have very simple texture components (ramps or tapering near edges) and can be coded without resorting to DCT texture coding techniques, but the general case will be dealt with by DCT or Shape Adaptive DCT described in the next section.

5.3.2 Shape-Adaptive DCT

Several techniques for spatial to frequency domain transformation of arbitrarily shaped regions have been reported in the literature [45]. The technique detailed here is a much lower complexity version that works within the standard 8x8 block coding structure while still performing well and was adopted by MPEG-4 video. Figure 5.9 and the following description from Sikora and Makai [46] outline the concept of the SA-DCT baseline algorithm for coding an arbitrarily shaped image segment contained within an 8x8-block.



(A) Original Segment

(B) Ordering of Pels and Vertical SA-DCT Used

(C) Location of Pels after Vertical SA-DCT

(D) Location of Pels Prior to Horizontal SA-DCT

(E) Ordering of Pels and Horizontal SA-DCT Used

(F) Location of 2-D SA-DCT Coefficients

Figure 5.9 : Successive steps involved for performing a SA-DCT forward transform on a segment of arbitrary shape.

The algorithm is based on predefined orthonormal sets of DCT basis functions. Figure 5.9 shows an example of an image block segmented into two regions, foreground (gray) and background (light). To perform the vertical transform of the foreground, the length (vector size *N*, $0<N<9$) of each column *j* $(0<j<9)$ of the foreground segment is calculated, and the columns are shifted and aligned to the upper border of the *8x8* reference block. Dependent on the vector size *N* of each particular column of the segment, a one-dimensional DCT with a transform kernel DCT-N containing a set of *N* basis vectors is selected for each particular column and applied to the first N pels of the column. Thus, the *N* vertical DCT-coefficients $\underline{c}_j$ for each segment column data $\underline{x}_j$ are calculated according to the formula:

$$\underline{c}_j = \sqrt{\frac{2}{N}} \cdot \underline{DCT - N} \cdot \underline{x}_j$$

$$with \quad \underline{DCT - N}(p,k) = c_0 \cdot \cos\left[p \cdot (k + 0.5) \cdot \frac{\pi}{N}\right] \quad and \quad c_0 = \sqrt{1/2} \quad if \quad p = 0; \quad c_0 = 1 \quad otherwise$$

$$for \quad 0 \le p,k \le N-1$$

For example, in fig. 5.9-B, the right most column is transformed using DCT-3 basis vectors. After SA-DCT in vertical direction, the lowest DCT-coefficients (DC values •) for the segment columns are found along the upper border of the *8x8* reference block (Figure 5.9-C). To perform the horizontal DCT transformation (Figure 5.9-E), the length of each row is calculated, and the rows are shifted to the left border of the 8x8 reference block, and a horizontal DCT adapted to the size of each row is then calculated using the above formulas. Note that horizontal SA-DCT-transformation is performed along vertical SA-DCT-coefficients with the same index (e.g. all vertical DC coefficients (•) are grouped together and are SA-DCT-transformed in horizontal dimension). Figure 5.9-F shows the final location of the resulting DCT coefficients within the *8x8*-image block.

In this algorithm, the final number of DCT-coefficients is identical to the number of pels contained in the image segment. Also, the coefficients are located in comparable positions as in a standard *8x8* block. The DC coefficient (•) is located in the upper left border of the reference block and, dependent on the actual shape of the segment, the remaining coefficients are concentrated around the DC coefficient. Note, that all positions of F(u,v) which are not defined after forward SA-DCT are explicitly set to zero. Since the contour of the segment is transmitted to the receiver prior to transmitting the MB information, the decoder can perform the shape-adapted inverse DCT as the reverse operation in both horizontal and vertical segment direction on basis of decoded BAB data:

$$\underline{x}_j = \sqrt{\frac{2}{N}} \cdot \underline{DCT - N}^T \cdot \underline{c}_j$$

Here $\underline{c}_j$ denotes a horizontal or vertical coefficient vector of size *N*.

5.4 Content Definition Problem

Although MPEG-4 fully supports techniques for coding shape and alpha transparency planes, the means to generate this information for the content to be coded is not specified. The generation of this so-called *segmentation* information is considered a producer or encoder issue. MPEG philosophy has always been that only decoder issues should be specified to guarantee inter-working; enabling competition between companies as to which can provide the best encoder engine, and also, in this instance, provide the most useful segmentation data. A large amount of producer material can be created now to take advantage of content based MPEG-4 coding. Blue-screen, or chroma keying techniques are commonly used in everyday weather forecasts and news bulletins and more often now in television special effects departments, and simple binary shape data is readily available from this source. Computer games or virtual reality applications should also allow easy object shape extraction from the predefined graphics models. The benefits of content based coding and manipulation will not however be readily realised for generic video. Content definition and extraction from generic video is an easy process for most of the animal kingdom. It is a very difficult task, though, for current generation computers, relying on complex image analysis and artificial intelligence techniques to perform even the most basic content analysis.

The video group of MPEG-4 has considered several automatic video segmentation algorithms which may well form an informative annex to the standard to help content producers. The next chapters detail the work of the author in development and investigation of several low level and data driven image segmentation techniques considered useful for this content definition problem. The *Analysis Model* will also be introduced as a rule based engine currently being defined by the COST 211^{quat} [20] group with the goal of true video content definition for MPEG-4.

6. Content Definition and Tracking by Feature Clustering

6.1 Background

The human visual system handles visual content definition and tracking in a seemingly effortless manner. Awareness of the immediate environment is a critical survival factor and nature has brought considerable resources to bear on the problem. The eye-brain mechanism itself makes use of upwards of 33 billion neurons devoted specifically to visual processing [47] and has had over 590 million years to evolve. Machine vision techniques, in contrast, remain very much in their infancy. Several computational techniques have been developed for certain content definition problems, typically within robot vision or medical image processing applications. Optical character recognition is possibly the most important commercial application. Magnetic resonance imagery, x-ray processing and fingerprint identification are other typical machine vision application domains in widespread use. In general, these techniques have been found to work well in only very structured lighting conditions or video from very controlled environments. The more ambitious target of automatic content definition from generic video, envisaged within the MPEG-4 coding community, has also recently been addressed but remains largely unsolved. This is the area tackled in this part of the research and the remainder of the thesis.

It is probable that our visual systems use a variety of visual cues or stimuli, e.g. brightness, colour, motion, depth, texture etc., to define what we see around us. It is also probable that these cues support a three dimensional internal model of the world which our brains develop in our formative years while we are moving around and interacting with our environment. Content definition for us can be viewed as a competition between bottom up processing of this raw visual data and top down model or expectation processing with a balance struck that hopefully gives us the correct information. It is not necessarily the case that visual stimuli are the dominant factor here. Interesting examples from [48] show how with people restored from blindness from birth to full sight later in life tactile information remains the most important bottom up stimulus even when the visual data is available. In fact subjects in this situation found that views of familiar objects remained just meaningless jumbles of colours and shapes until they could pick the objects up and touch them.

Bottom-up visual data processing has been investigated, in this research, and an algorithm that allows an arbitrary number of visual cues to be used has been developed. Feature vectors made up of brightness, colour, motion and position data for each digitised pixel have been constructed and input into a multidimensional clustering algorithm described in section 6.2. Clusters are then mapped to a spatial segmentation on a frame by frame basis and results are presented in section 6.3. Many of the variables associated with the clustering procedure have been investigated with respect to the segmentation application and conclusions drawn. Section 6.4 details the extension of the clustering segmentation work to cluster tracking in subsequent video frames and shows good primary content of interest (i.e. foreground / background) extraction results. Section 6.5 concludes the clustering segmentation and tracking work.

6.2 Feature Clustering

A good definition of clusters for our purpose, from [49] and [50], is "connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points". The multi-dimensional space in the image feature context can be constructed from a set of individual pixel properties (e.g. luminance, colour, position), a set of neighbourhood properties (e.g. texture, motion), or any combination. Figure 6.1 shows one such combination of pixel attributes, Y representing luminance and U and V representing the two colour difference signals. This colour space is used commonly within digital source video and its derivation from the more usual RGB space is defined in ITU-R Rec. BT.601 [1]. The next subsections describe the clustering technique used and its application to partition the image feature space. Different factors affecting the clustering process are also further examined.

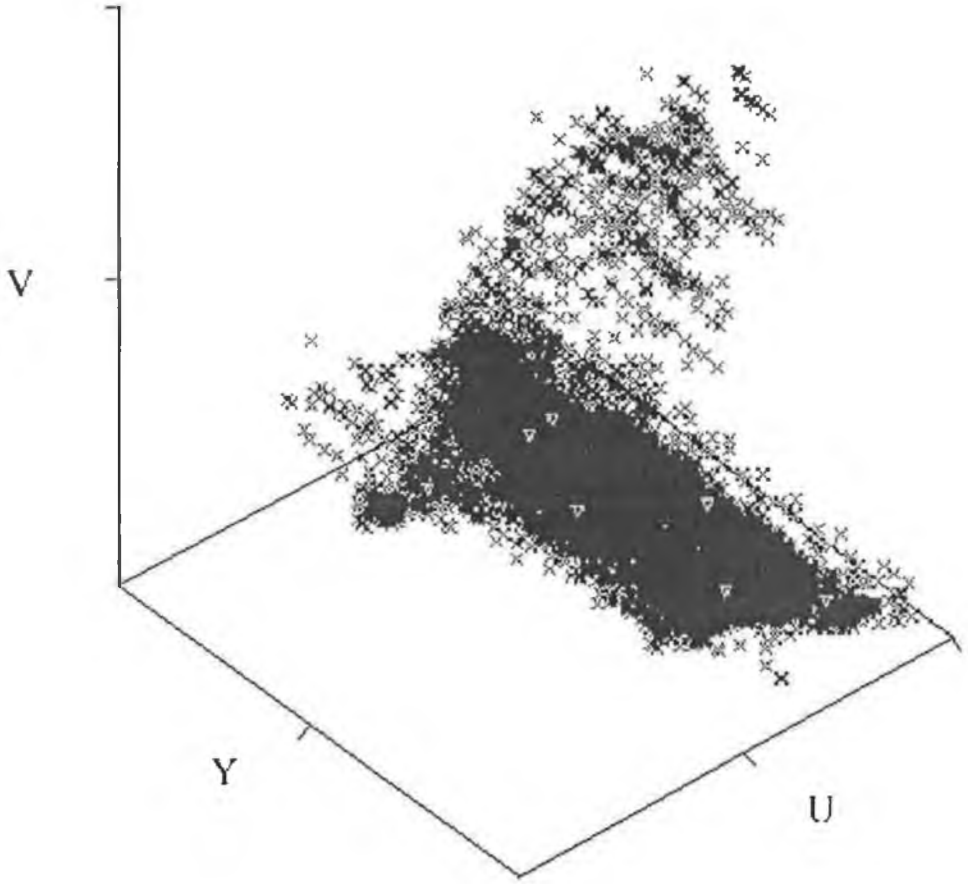


Fig 6.1 : {y, u, v} Feature space of 'Foreman' frame 0. Feature vectors represented as points in 3-dimensional YUV space.

6.2.1 Clustering Procedure using the K-means Algorithm

The clustering procedure used here is based on the 'K-means' algorithm detailed by Tou and Gonzalez [50]. The K-means algorithm is a popular clustering technique used in many applications, including the initialisation of more computationally expensive algorithms (Gaussian mixtures, Learned Vector Quantisation, Hidden Markov Models). Nowlan [83] and Bottou et al. [84] refer to K-means type algorithms as hard threshold techniques and compare to the soft threshold approaches of Guassian Mixtures and Baum Welch. The latter are often preferred because they have an elegant probabilistic framework and a general optimisation algorithm named EM (Expectation-Maximisation) [85]. Bottou and Bengio [84] actually describe the K-means algorithm by extending the mathematics of the EM algorithm to the hard threshold case and show that K-means, in addition to fast convergence properties, actually outperforms Gaussian mixture EM in terms of feature space quantisation error minimisation. EM using the mixture of Gaussian approach had already been shown to approximate the Newton optimisation algorithm [86]. K-means, however is proven to minimise quantisation error using exactly the Newton algorithm.

In the area of image segmentation K-means clustering is popular in texture segmentation problems [80] as the technique easily extends to the high dimensionality of texture moment feature vectors. It is also popular in biomedical image segmentation [87] since the number of clusters (K) is usually known for images of particular regions of human anatomy. The latter requirement for *a priori* knowledge is also a problem, however, for our goal of an unsupervised yet generally applicable video segmentation tool. The approach adopted here for this research has been to set K to a larger than anticipated number of region clusters and then define a merge threshold of inter-cluster distances. This kind of merging process is very heuristic but its merit lies in the fact that it is efficient and requires a minimum of human interaction.

The algorithm developed here is initialised with *k* arbitrary clusters and is designed to iteratively minimise the distance from all points in each cluster domain to the cluster centre. Each feature vector is allowed to change allegiance between clusters once per iteration but clusters within a proximity threshold of each other can also be merged. Both Euclidean and Mahalanobis distance metrics, described in section 6.2.2, have been used. The resulting clusters can then be mapped to spatial segmentations of the scene into hopefully useful content regions. The procedure consists of the following steps:

1. Form initial clusters from simple connected regions (stripes or blocks) of the image frame data.
2. Calculate the distance of each vector to each cluster centre and reassign the vector to the closest one.
3. Calculate inter-cluster distances and merge clusters within a certain threshold.
4. Repeat steps 2-3 until the number of reassigned features drops below a small threshold (1% of all features in these tests) or an iteration limit is reached.

A labelled region image, a mean valued {y, u, v} image and statistics of the remaining clusters form the result set for each frame.

The behaviour of the k-means algorithm is influenced by many factors: the number and choice of initial cluster centres, the order in which features are reassigned, the distance metric used, the proximity thresholds and the scale and geometric properties of the feature vector space. Section 6.3 investigates several of these factors applied to video sequences taken from the COST 211ter and ISO/MPEG-4 test data set.

6.2.2 Distance Measures

Different metrics can be used to measure dissimilarity of feature points to existing clusters. The most popular measures are the Minkowski metric family:

$$d(i,k) = \left(\sum_{j=1}^{d} \left| x_{ij} - x_{kj} \right|^{r} \right)^{1/r} \quad , \; r \geq 1$$

If r = 1 we have the *Manhattan* or *City-block* distance. With binary features this is also known as the *Hamming* distance. If r = 2 we have the more familiar *Euclidean* distance measure which is invariant to translations and rotations in the pattern space.

$$d(i,k) = \sqrt{\sum_{j=1}^{d} \left| x_{ij} - x_{kj} \right|^{2}} = \left[(\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_i - \mathbf{x}_k) \right]^{1/2}$$

Normalisation is generally desirable when using this metric. Otherwise, one feature (e.g. luminance) can dominate in the distance calculation. A process of normalisation by re-scaling was carried out on the sample set. Normalisation followed by another scaling was also used in some tests to change the importance of some feature components (e.g. position).

A third measure known as the Mahalanobis distance was also used in this clustering work. This measure incorporates feature correlation information in the form of the inverse pooled sample covariance matrix. The effect of this is to standardise each feature to zero mean and unit variance. A normalisation process is then not required with this metric as all feature vector components are assigned equal importance.

$$d(i,k) = \left[(\mathbf{x}_i - \mathbf{x}_k)^T C^{-1} (\mathbf{x}_i - \mathbf{x}_k) \right]^{1/2}$$

C^{-1} was estimated in this work by calculating the covariance matrix directly, then applying LU decomposition and back-substitution to determine the inverse. If C is the identity matrix this metric also reduces to be identical to the Euclidean distance.

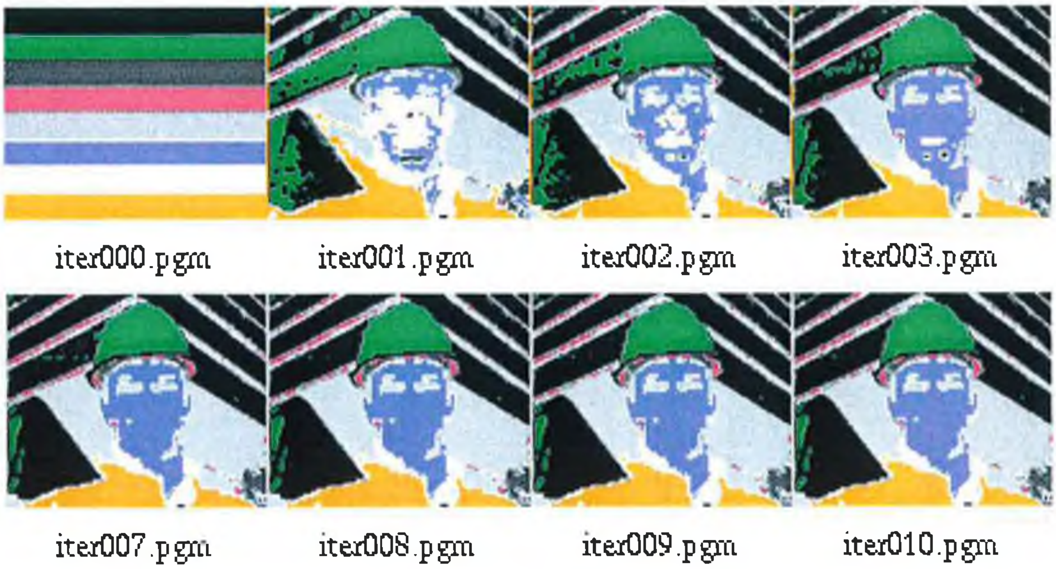
6.3 Clustering Results

The first set of results show the iterations of the clustering process itself from an initial block or stripe initialisation through to convergence on frame zero of the test material. Spatially connected clusters were chosen deliberately to initialise the clustering process since the hope was that final cluster results would also exhibit this characteristic. For each initial condition, the different Mahalanobis and Euclidean metrics were also tested.

6.3.1 Colour Clustering - {y, u , v} feature space

6.3.1.1 Stripe Initialisation

The following figures 6.2 and 6.3 show Mahalanobis and Euclidean metric results respectively. Pseudo colour has been added to both figures to highlight individual clusters at each iteration.

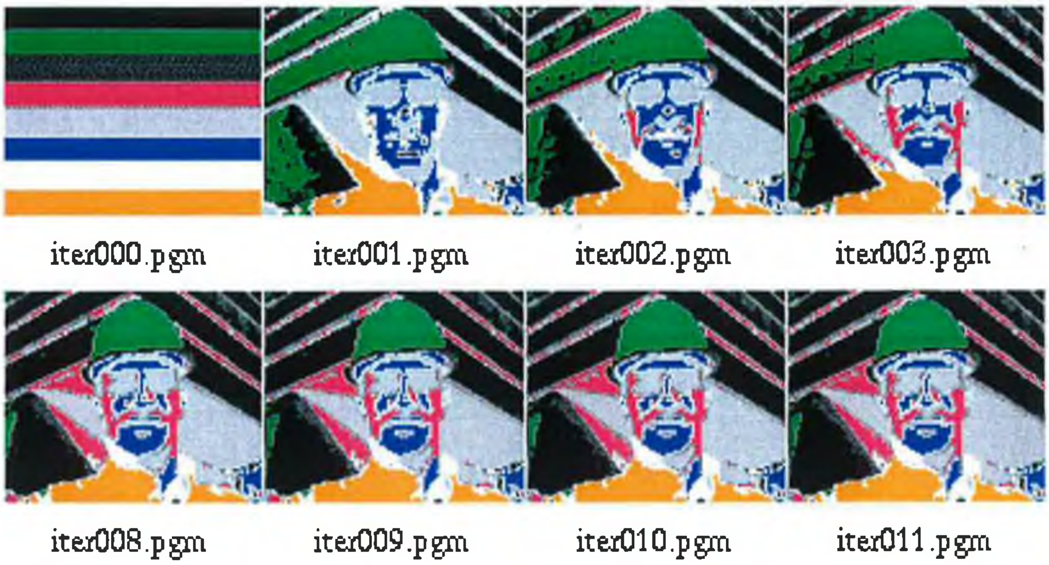


iter000.pgm iter001.pgm iter002.pgm iter003.pgm

iter007.pgm iter008.pgm iter009.pgm iter010.pgm

Figure 6.2 : Clustering process using pooled sample Mahalanobis metric. Sample data was {y,u,v} feature vectors of QCIF 4:2:0, frame 0 of 'Foreman'.

Figure 6.2 shows an 8-cluster initialisation with Mahalanobis distance clustering. The optimal distance threshold was found by experiment to be 0.6. This process required 10 iterations before completion and algorithm speed is approximately 2 iterations per second on an UltraSparc-1 platform and on this data set.

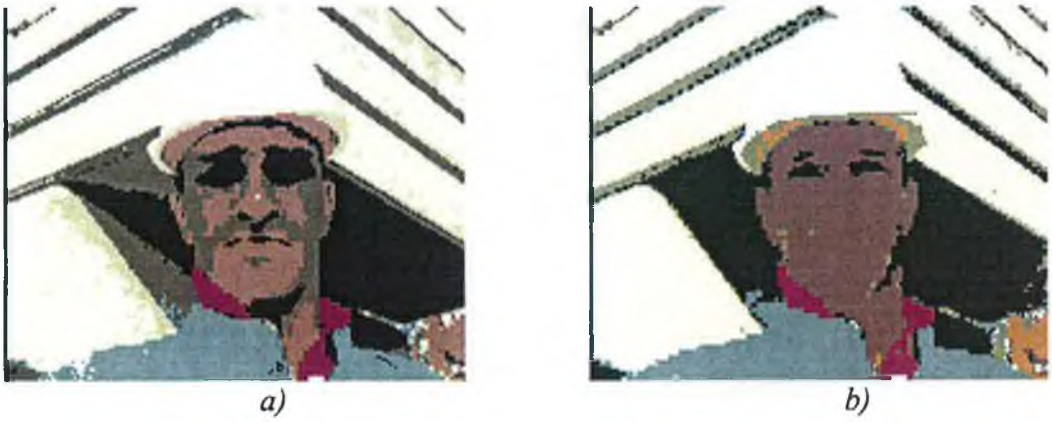


iter000.pgm iter001.pgm iter002.pgm iter003.pgm

iter008.pgm iter009.pgm iter010.pgm iter011.pgm

Figure 6.3 : Clustering process using the Euclidean metric. Sample data was {y,u,v} feature vectors of QCIF 4:2:0, frame 0 of 'Foreman'.

Figure 6.3 also shows a stripe initialisation (i.e. 8 initial clusters) and K-means Euclidean distance clustering with a distance threshold, in this case, of 0.11. The algorithm requires 11 iterations before completion. Again the iteration rate is approximately 2 per second on the UltraSparc platform.



a) *b)*

Figure 6.4 : Mean value regional images of stripe initialised clustering of 'Foreman' a) Euclidean and b) Mahalanobis distance metrics

Figure 6.4 shows mean value region images of the resulting partitions and highlights the difference observed when using the different distance metrics. The Euclidean clustering normalises components by re-scaling to a common range but this ignores the shape of the distributions. The luminance distribution is more skewed towards the top of the range and, hence, luminance is the more prominent feature in the resulting segmentation. The Mahalanobis result, on the other hand, standardises each component to zero mean and unit variance. No single feature dominates in this case.

Figure 6.5 gives an indication of the speed with which relatively stable clusters are formed with the clustering process and this was a typical result for all clustering tests.

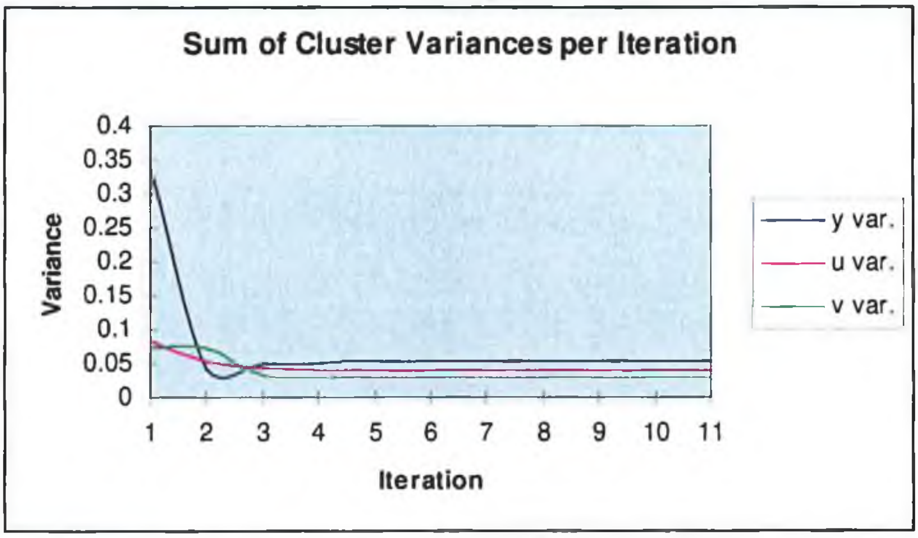


Figure 6.5 : Sum of cluster variances per iteration for Euclidean clustering of Foreman, frame 0, showing fast algorithm convergence.

By iteration 4, variance levels have converged to very close to their final levels. This can also be seen in the spatial cluster mappings per iteration of fig. 6.2. Iterations could be stopped at this point with little penalty. The halt mechanism in this particular process was related to the number of reassigned features dropping below a pre-defined limit, i.e. 1% of the entire feature space for these tests. A second observation was that the final clusters were not critically dependent on the initial seed clusters. Figure 6.6 below shows almost identical cluster label images resulting from horizontal and vertical stripe initialisation. This indicates a stable minimum, if not the global minimum, of the feature space quantisation error has been reached in this case.

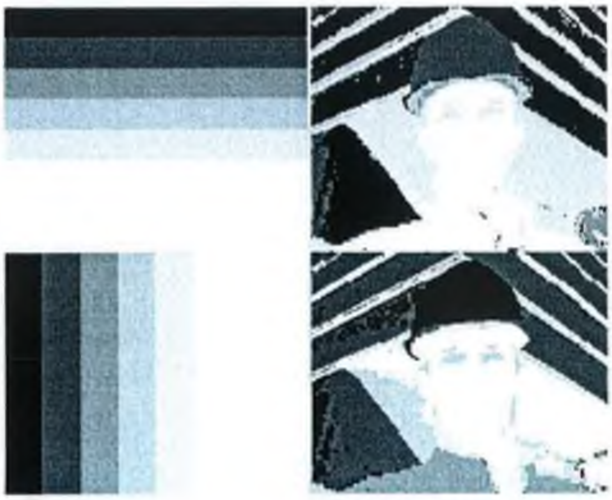
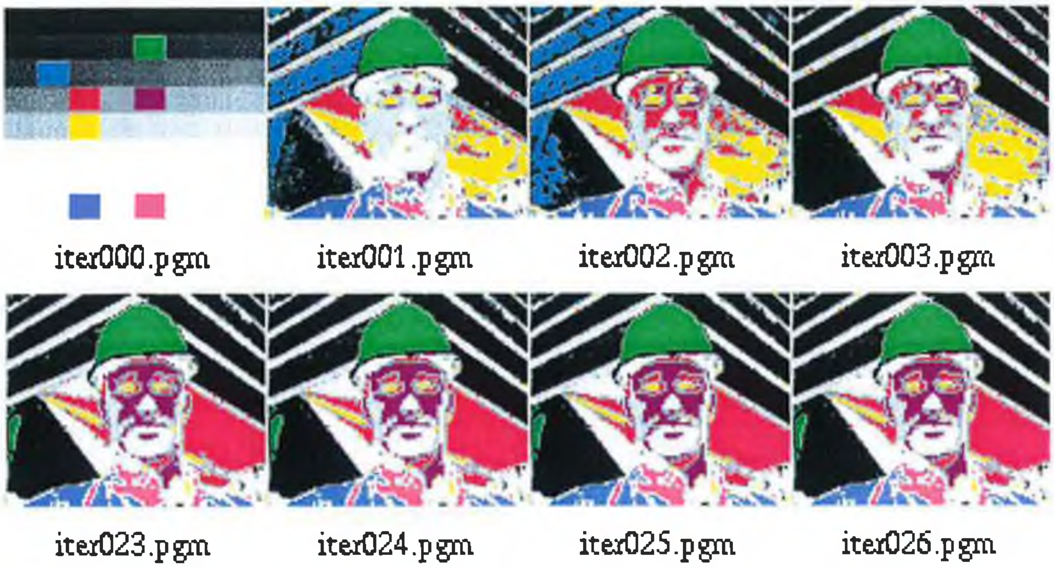


Figure 6.6 : Horizontal and Vertical stripe initialisation and resulting cluster regions

6.3.2 Block Initialisation

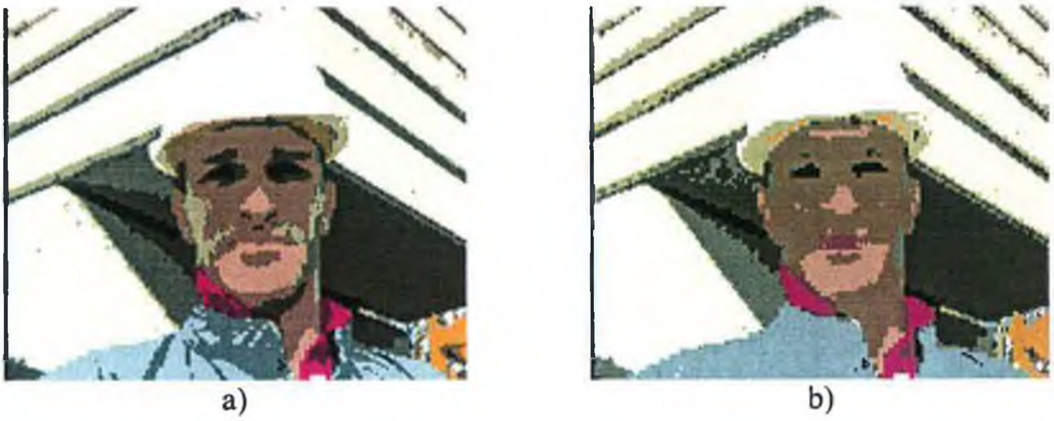
The same test frame is shown here in the following result set for comparison. A larger number of initial cluster centres (K=64) was tried in these tests.


iter000.pgm iter001.pgm iter002.pgm iter003.pgm

iter023.pgm iter024.pgm iter025.pgm iter026.pgm

Figure 6.7 : Clustering process using Euclidean distance metric. {y,u,v} feature vectors of QCIF 4:2:0 frame 0 of 'Foreman' sequence, block initialisation (64 clusters), distance threshold = 0.11, 26 iterations

Once again pseudo-colour has been added, to both Euclidean clustering results of figure 6.7 and Mahalanobis results of figure 6.9, to highlight several of the most important clusters through all iterations. Figure 6.8 shows mean value region images of the resulting segmentations.


a)

b)

Fig 6.8 : Mean value images of final clustering result. a) Euclidean, 14 final clusters and b) Mahalanobis, 12 final clusters

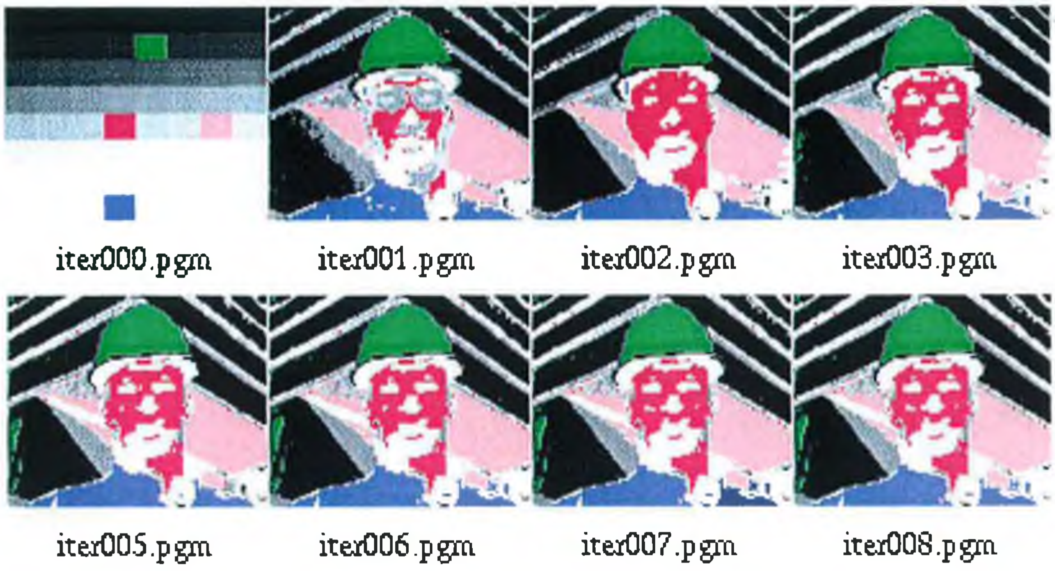


Figure 6.9 : Clustering process using pooled sample Mahalanobis metric. {y,u,v} feature vectors of QCIF 4:2:0 frame 0 of 'Foreman' sequence, block initialisation (64 clusters), distance threshold : 0.6, 8 iterations

The high number of initial clusters, K = 64, is quickly reduced to a stable distribution of 14 or 12 clusters depending on distance measure and respective thresholds. The clustering strategy, merging clusters within a specified threshold, works well in this case but the optimal threshold value must be determined manually. Once determined for this feature distribution, however, the same value can be used in subsequent frames to yield acceptable segmentations.

6.3.3 Incorporating Motion Data

This section, and [54,55], details work to incorporate pixel level motion information into the feature space. A coarse-to-fine optic flow algorithm [78] was used to derive the x- and y- optic flow images for a frame pair within the 'foreman' sequence. Figure 6.10 shows two temporally adjacent frames of the sequence.

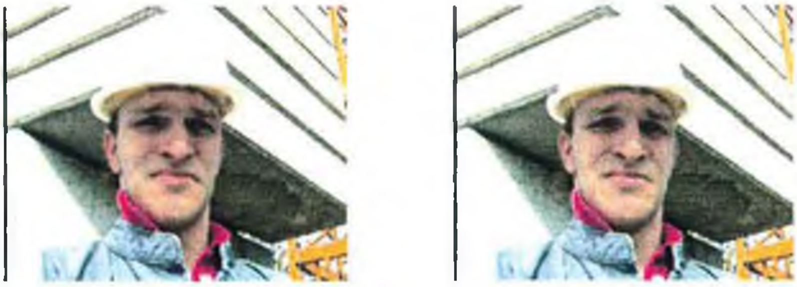
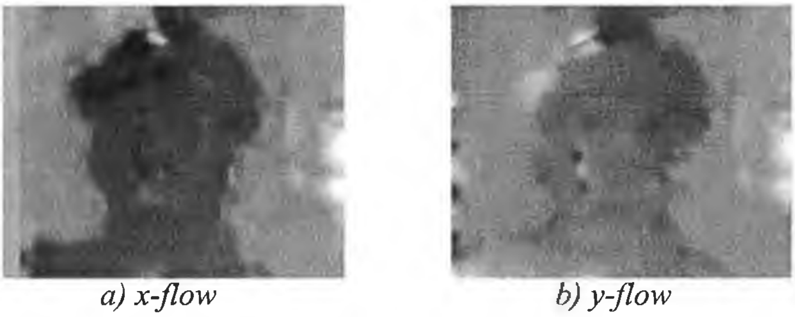


Figure 6.10 : Frame pair from 'Foreman' sequence

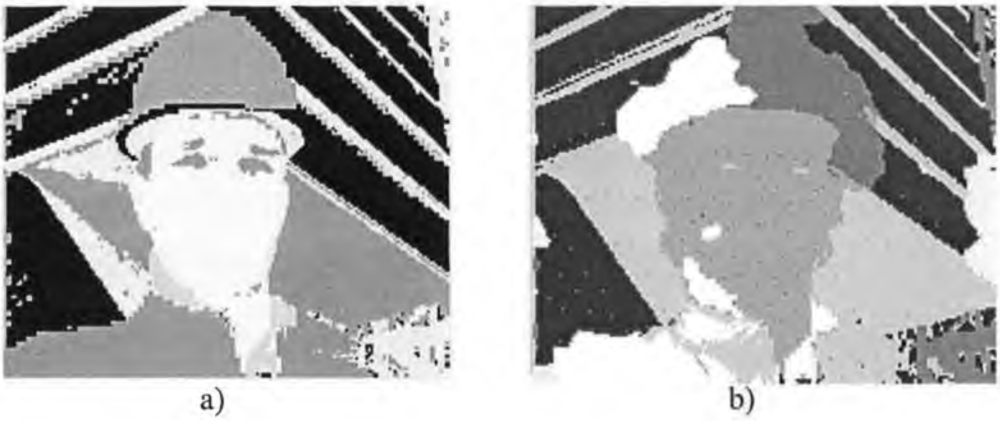
Figure 6.11 a) and b) shows the resulting x and y flow images. For these flow images, mid-level grey represents zero pixel motion and black and white areas represent the motion extremes.


a) x-flow

b) y-flow

Figure 6.11 : Resulting optic flow images.

This flow information was initially used directly in the feature clustering process using 5 dimensional feature vectors of {y, u, v, x-flow, y-flow}. Figure 6.12 b) compares the Mahalanobis clustering result of this information with colour only clustering of figure 6.12 a).

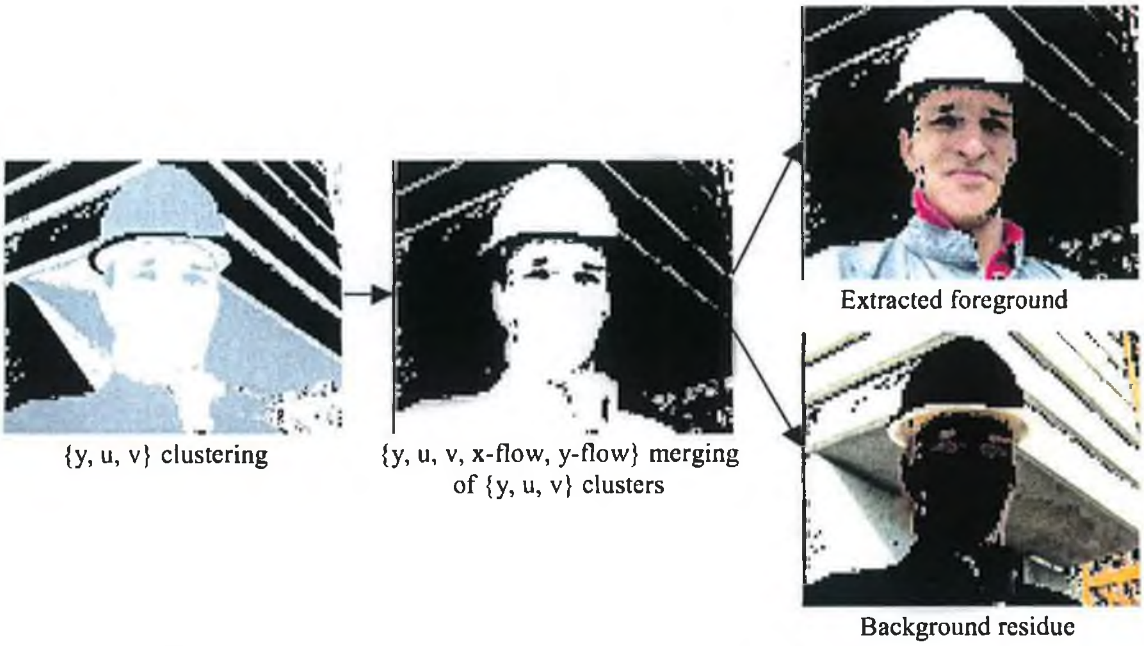

a)

b)

Figure 6.12 : Clustering results for a) {y, u, v} and , b) {y, u, v, x-flow, y-flow} feature vectors. Mahalanobis clustering, 8 stripe initialisation.

Two observations can be made. Boundary precision of the colour only clustering has been lost due to inaccuracies in optic flow data at region boundaries but final regions are generally more coherent in space due to smoothness constraint of motion data. The influence of the optic flow data could have been reduced by clustering with scaled data and using the Euclidean metric, however, a more useful result was obtained by first clustering colour only data then merging clusters based on inter-cluster distance in the combined colour and motion feature space.

Figure 6.13 below illustrates the process of clustering initially with {y, u, v} feature data, then, merging clusters based on new cluster centers (average {y, u, v, x-flow, y-flow}).



{y, u, v} clustering

{y, u, v, x-flow, y-flow} merging of {y, u, v} clusters

Extracted foreground

Background residue

Figure 6.13 : Foreground / Background extraction by incorporation of motion information.

This result shows good, but noisy, foreground and background extraction for this data set. However, extension of this procedure for the whole sequence makes less sense. Figure 6.13 works since, at that instant, all regions of the foreman are exhibiting common motion. When the foreman stops moving relative to the background the distinction can not be made with motion criteria alone. Figure 6.14 illustrates the problem with subsequent frames of the sequence.

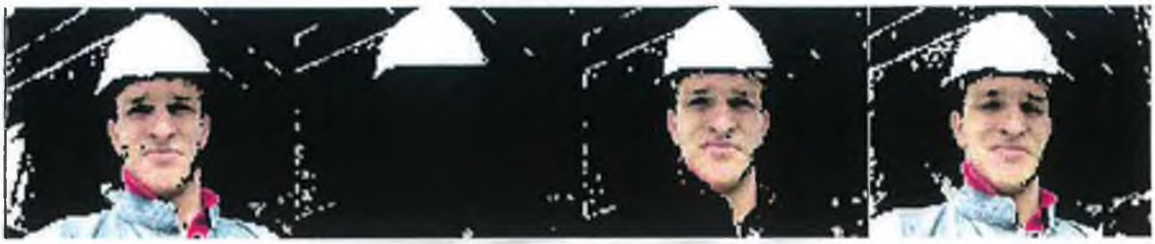


Figure 6.14 : Instantaneous common motion foreground regions

Another interesting observation here concerns the general reliability and accuracy of the optic flow data. Flow data is generally more reliable in highly textured areas. The smooth uniform texture of the foremans hat results in erroneous motion estimates which in the case of figure 6.14 has been mistaken for foreground motion. In fact the hat is still for these frames.

Motion information has not been investigated further in this chapter, however, it is one of the most important visual cues and all the analysis frameworks of chapter 7 rightly integrate motion segmentation information. Regions coherent in both spatial and temporal attributes are still considered the most useful in terms of content extraction, however, the spatial accuracy of the regions, and the incorporation of positional data, is the focus of the next sections.

6.3.4 Colour and Position Clustering

The following result set, figures 6.15 to 6.22, shows the effect of adding positional information to the feature space (i.e. using 5 dimensional feature vectors of {y, u, v, x-coord, y-coord}). The Mahalanobis distance metric assigns equal importance to all features but better results can be achieved by using Euclidean distance within a normalised space and allowing the positional information to be scaled. Again the selection of the best scale factor must be supervised.

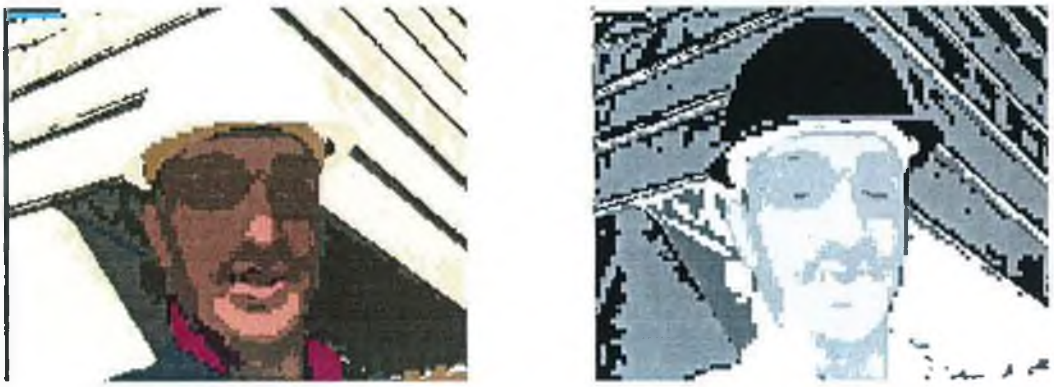


Figure 6.15 : {y, u, v} clusters, a) mean valued and b) region label images. Mahalanobis distance, block initialisation, t=0.6, 13 iterations, 13 bins

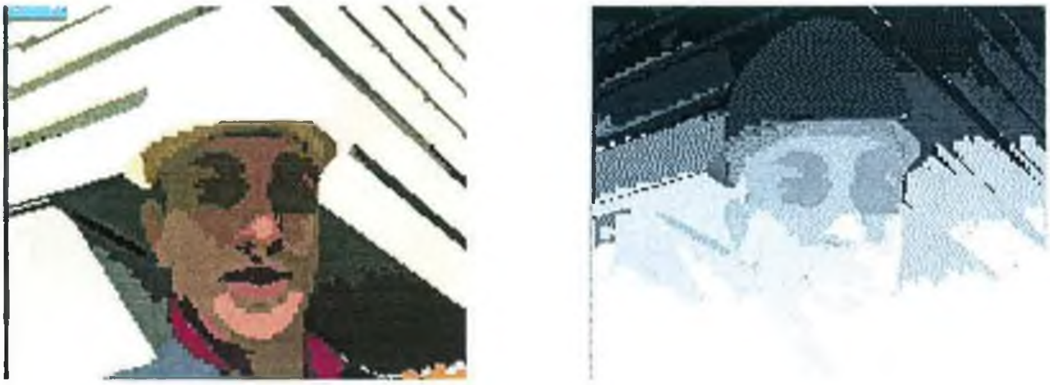


Figure 6.16 : {y, u, v, x-coord, y-coord} clusters, a) mean valued, b) region label images. Mahalanobis distance, block initialisation, t=0.6, 13 iterations, 37 bins

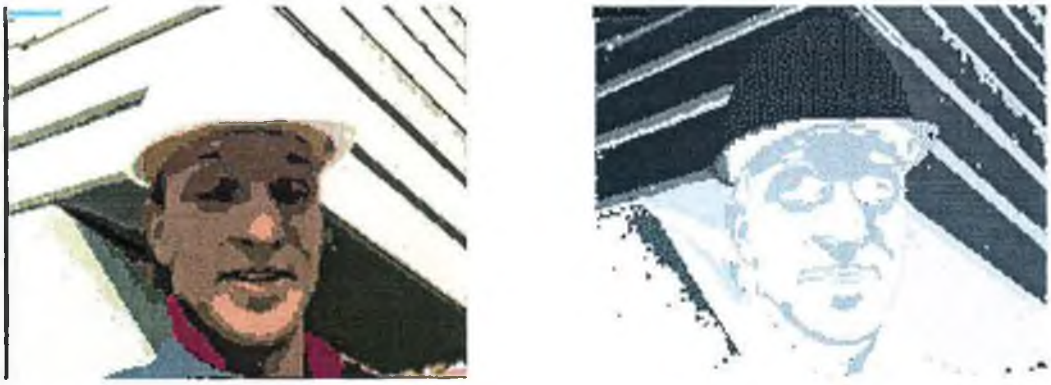


Figure 6.17 : {y, u, v, x-coord, y-coord} clusters, a) mean valued, b) region label images. Euclidean distance, x,y-coords scaled to 20%, t=0.1, 11 iterations, 19 bins

The effect of positional information is clearly seen in this result set. The numerous, small and disconnected regions of figure 6.15 are effectively merged, with the extra spatial data, into the more homogeneous regions of figure 6.16. Many arbitrary boundaries are also introduced, however, in the Mahalanobis result which is giving the *x* and *y* co-ordinate data equal importance. The result of figure 6.17 lessens the importance of the co-ordinate data by re-scaling values to 20% of original range. Resulting regions are more generally homogeneous than figure 6.15 and parts of the building and the foreman's hat remain clearly separated, yet there are less arbitrary boundaries than in figure 6.16.

The following result set, figures 6.18 to 6.22, show the same effect of spatial data on cluster segmentations of another QCIF image frame from the *Mother and Daughter* test sequence and a CIF frame from the *Stefan* sequence.

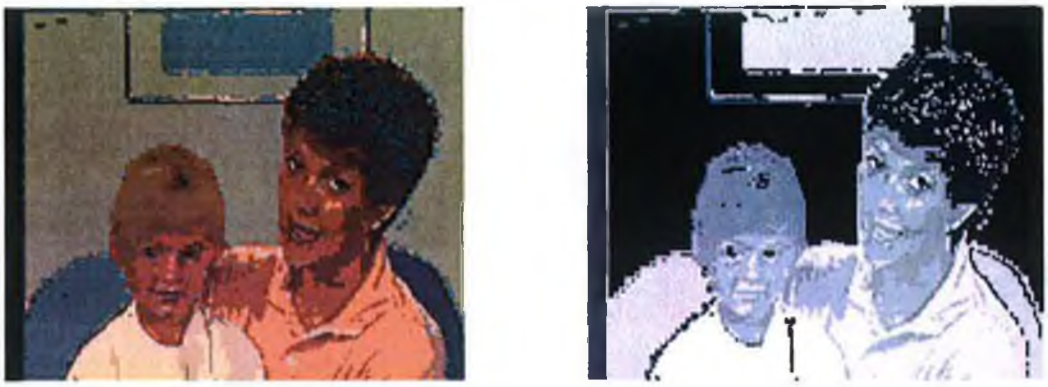


Figure 6.18 : {y, u, v} clusters, a) mean valued and b) region label images. Mahalanobis distance, block initialisation, t=0.7, 9 iterations, 15 bins

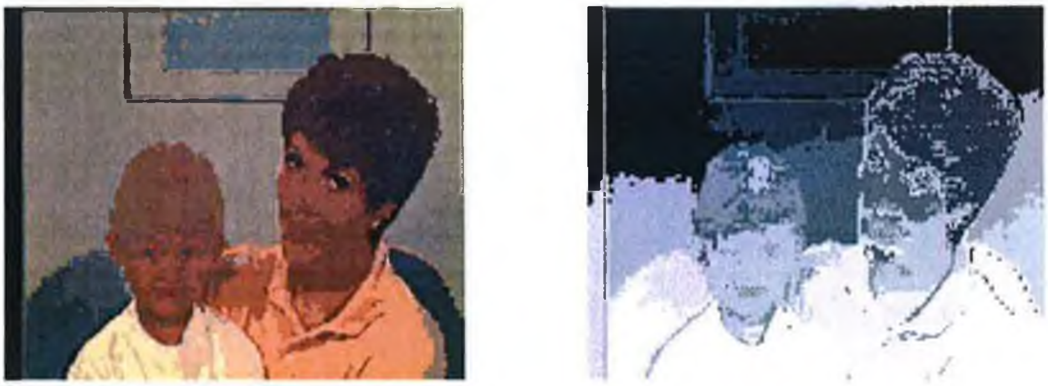


Figure 6.19 : {y, u, v, x-coord, y-coord} clusters, a) mean valued, b) region label images. Mahalanobis distance, block initialisation, t=0.7, 15 iterations, 39 bins

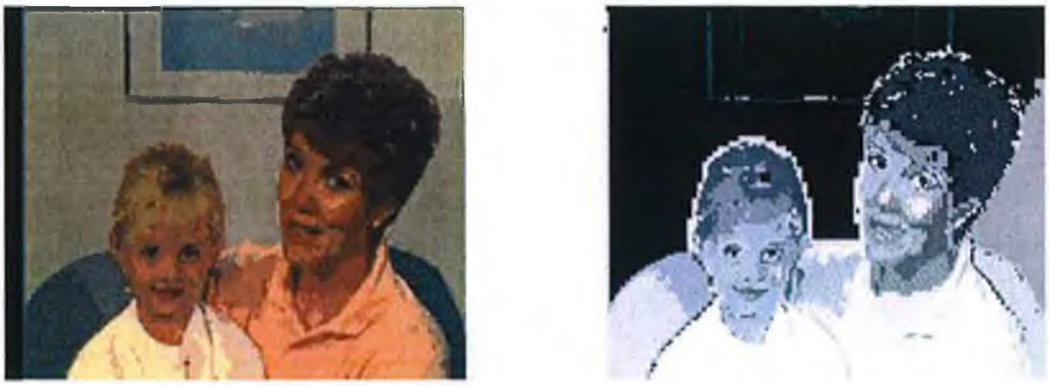


Figure 6.20 : {y, u, v, x-coord, y-coord} clusters, a) mean valued, b) region label images. Euclidean distance, x,y-coords scaled to 20%, t=0.1, 12 iterations, 37 bins

Figures 6.21 and 6.22 show clustering results for the larger CIF (352 x 288) source frame from Stefan.

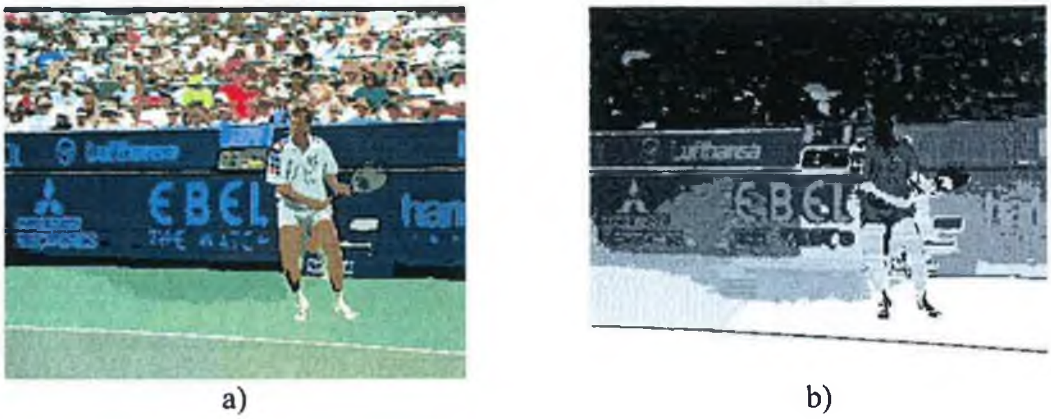

a) b)

Figure 6.21 : a) Mean valued and b) region label segmentation images of CIF 4:2:0, Stefan frame 0. Mahalanobis distance metric, 64 initial clusters, 48 final cluster bins

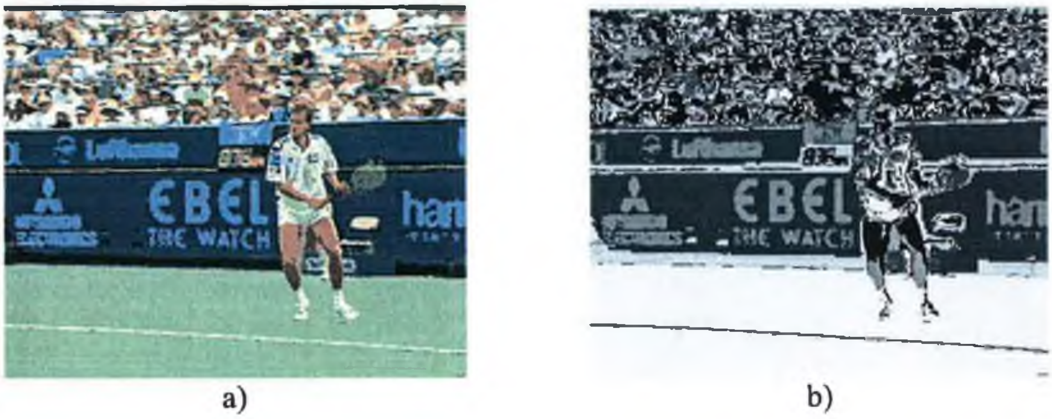

a) b)

Figure 6.22 : a) Mean valued and b) region label segmentation images of CIF 4:2:0, Stefan frame 0. Euclidean distance metric, x,y - coordinates scaled to 10%, 64 initial clusters, 13 final cluster bins

Again the segmentations here of the background advertisement boards and the tennis court surface of this frame show good spatial homogeneity in both clustering results. The scaled Euclidean case does have less arbitrary boundaries, however.

As a general observation for this section, spatial homogeneity has been effectively introduced into the clustering process by the addition of the spatial co-ordinate data. Our final clustering vector set is comprised of simpler than expected 5-component vectors. The next sections detail work in post-processing of the final segmentations of both colour only and colour and spatial data clustering and also tracking of regions through the entire sequence. A tracking framework for content-of-interest extraction for generic video through feature clustering is proposed and tested.

6.3.5 Morphological Post-processing

It is only in the feature space incorporating position data that the property of spatial connectivity has an influence to the final result. The colour-only clustering results impose no specific connectivity constraint and yet colour cluster mappings do largely display this property. The region boundary images, however, do show how useful content boundaries are accompanied by a lot of spurious and unwanted detail. Fig. 6.23 a) shows many small, even singleton, disconnected regions. Results from morphological processing techniques to eliminate these unwanted regions are presented in this section. Morphology is a key segmentation tool in its own right and details of morphological processing are presented in the next chapter.

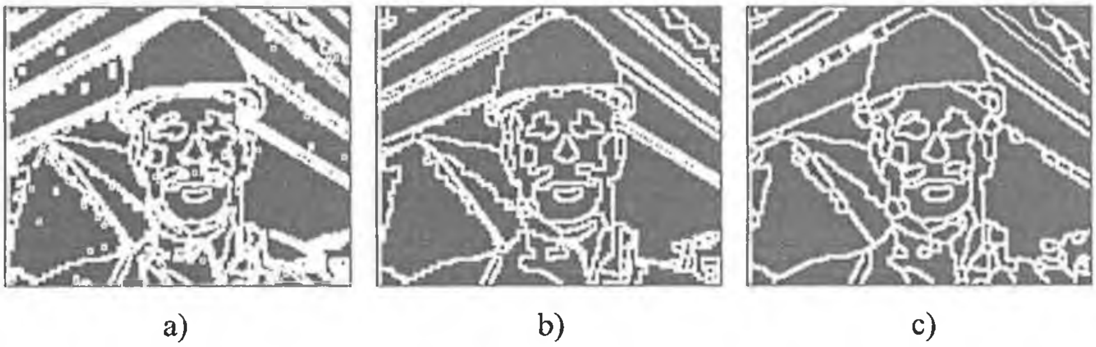
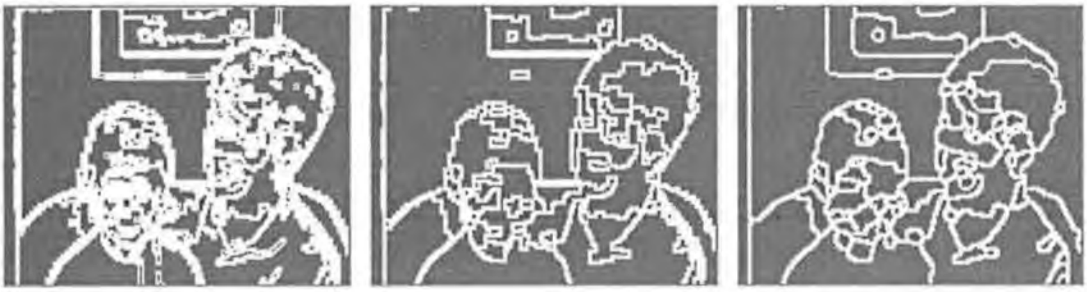


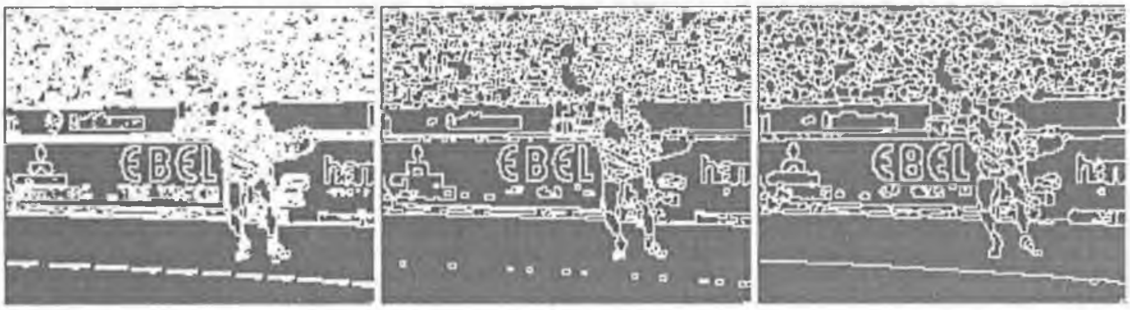
Fig. 6.23 : Morphological post-processing applied to clustering segmentation maps a) original mapping boundary image, b) effect of erosion/dilation, c) application of watershed filtering. (14 clusters bins but 135 watershed connected regions)

Fig. 6.23 above shows region boundary images of the unprocessed cluster mapping, and with simple open-close morphological filtering [51] and the greyscale watershed, described in the next chapter, applied respectively. The morphological filtering for fig. 6.23 b) is in fact a morphological erosion followed by a dilation and both filters use a 3x3 structuring element for this case. Figures 6.24 and 6.25 show similar simplified boundary images for Mother and Daughter and Stefan.



a) b) c)

Fig. 6.24 : Morphological post-processing applied to 'mother + daughter' clustering segmentation a) original mapping boundary image, b) effect of erosion/dilation, c) application of watershed filtering. (15 clusters bins, 94 watershed connected regions)



a) b) c)

Fig. 6.25 : Morphological post-processing applied to 'Stefan' clustering segmentation a) original mapping boundary image, b) effect of erosion/dilation, c) application of watershed filtering. (13 clusters bins but 764 watershed connected regions)

Figures 6.24 c) and 6.25 c) show how important content edges of the picture frame and the tennis court boundary line respectively are not retained in the simple erosion/dilation filtering but are preserved in the watershed. More details of watershed morphology are presented in the next chapter but even the simple erosion / dilation morphological operations have been shown to improve on the final content definition segmentation maps.

6.4 Tracking Content of Interest

The clustering and content definition process helped partition each video frame into useful content regions but these must now be combined and tracked in some way to extract video content. In the following test results the initial clustering output of the first sequence frame was examined and used to manually determine which region belonged to the content of interest and which did not. Subsequent frames were then compared to this base set and tracked. The following steps were used in the tracking process:

- The region cluster mean vectors of the content to be tracked, within the first frame the content appears in, were determined and stored.
- Subsequent region cluster means are then compared to this set and the closest ones deemed content of interest (COI).
- Clustering process may result in more or less region clusters emerging in COI than in original set. Two thresholds T_{COI-1} and T_{COI-2} were used to reject closest but distant regions and accept next closest and nearby regions respectively.
- Closest matching regions to the previous vector set form the vector set for the next frame.

Some justification for the thresholds used in this tracking framework is given in the next section. Results are then presented, in the following sections, of tracking three ITU-T full length video test sequences, *Mother and Daughter*, *Salesman* and *Foreman*.

6.4.1 Content Extraction Thresholds

The heuristic parameter set has now grown to four variables; the cluster merge threshold, the component scaling factor and the two content-of-interest tracking thresholds and this needs some justification.

The cluster merge threshold is a function of the feature space and the distance metric used and can be determined quickly for the first frame of a sequence to be segmented by performing a small number of trials and observing results on that single frame. It allows us to avoid specifying *K*, the number of final clusters to form, and was found to work well in the previous single frame segmentation tests.

The component scaling factor allows us to change the importance of certain components in our feature vector. It is used, here, to change the importance of the positional information to between one and two fifths of their original QCIF or CIF range. This can again be determined quickly usually on just the first frame of the sequence to give the optimum balance between spatially homogeneous regions and arbitrary boundaries between these regions.

The two content-of-interest (COI) thresholds are introduced here to provide a mechanism to determine which clustered regions of frame N should be retained in the set of COI regions of frame N-1. If regions are no longer present in the latest frame we do not want background or other non-COI regions labelled incorrectly, hence T_{COI-1}. Similarly if a region splits into several separate clusters in the subsequent frame but all these clusters remain close in feature space, we would like to include these new clusters as part of our new COI set. T_{COI-2} allows us to accept these regions.

These variables do make this a highly heuristic process, however, they can all be quickly estimated with just a few frames of the data to be processed and the following results applied over many hundreds of frames show how efficient the process can be.

6.4.2 Mother & Daughter Result

The best result here was obtained by tracking the 20 background clusters detailed in table 6.1. These were selected manually by mouse click over the segmented first frame. Throughout the tracking process the Euclidean distance metric was used with x,y - coordinate data scaled to 40%. An acceptable value for both T_{COI-1} and T_{COI-2} was found to be 0.083 in Euclidean space. Cluster merge threshold was set to 0.1. The content of interest was then, as shown in figure 6.26, everything but these tracked regions.

Table 6.1 : Background cluster descriptions and feature vector means for Mother & Daughter (frame 0)

Cluster	Content	{y, u, v, x-coord, y-coord} region means
0	pt of left stripe (sml)	{0.020368, 0.737891, 0.189787, 0.013734, 0.159464}
1	btm pt of picture	{0.482697, 0.767640, 0.054580, 0.195702, 0.045482}
2	mid pt of picture	{0.494490, 0.670131, 0.138509, 0.220817, 0.027449}
3	top right corner	{0.606837, 0.437672, 0.287798, 0.357079, 0.041479}
4	Picture frame edge (sml)	{0.294851, 0.552067, 0.212880, 0.131572, 0.036377}
5	top left corner	{0.597619, 0.440627, 0.216173, 0.074296, 0.067737}
6	left pt picture frame	{0.633812, 0.542978, 0.189405, 0.132528, 0.051155}
7	rest of picture	{0.544704, 0.560397, 0.254921, 0.242077, 0.025825}
10	mid pt back wall	{0.618744, 0.429631, 0.302437, 0.197183, 0.146352}
12	rest of picture edge (sml)	{0.351685, 0.501288, 0.320937, 0.334017, 0.099291}
13	pt of left stripe (sml)	{0.494782, 0.736788, 0.185116, 0.016000, 0.116084}
14	left pt back wall	{0.600771, 0.435016, 0.255869, 0.055851, 0.204122}
22	right pt back wall	{0.560352, 0.483580, 0.281327, 0.376302, 0.207728}
23	edge left of child	{0.428784, 0.565636, 0.222705, 0.084221, 0.323032}
24	pt of chair left of child	{0.455293, 0.706472, 0.159146, 0.071304, 0.294628}
29	pt of chair right of mother	{0.309800, 0.816205, 0.121860, 0.375474, 0.321429}
30	top half left stripe	{0.024640, 0.981485, 0.128205, 0.005714, 0.102098}
31	rest of chair left of child	{0.391485, 0.767797, 0.085777, 0.044416, 0.343972}
36	rest of chair right of mother	{0.527363, 0.557209, 0.255495, 0.384190, 0.301687}
37	Bottom half left stripe	{0.024654, 0.975449, 0.126946, 0.006214, 0.307168}

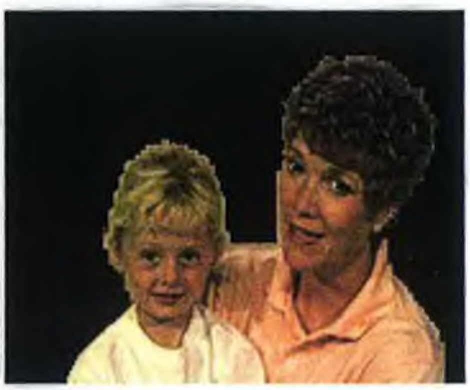
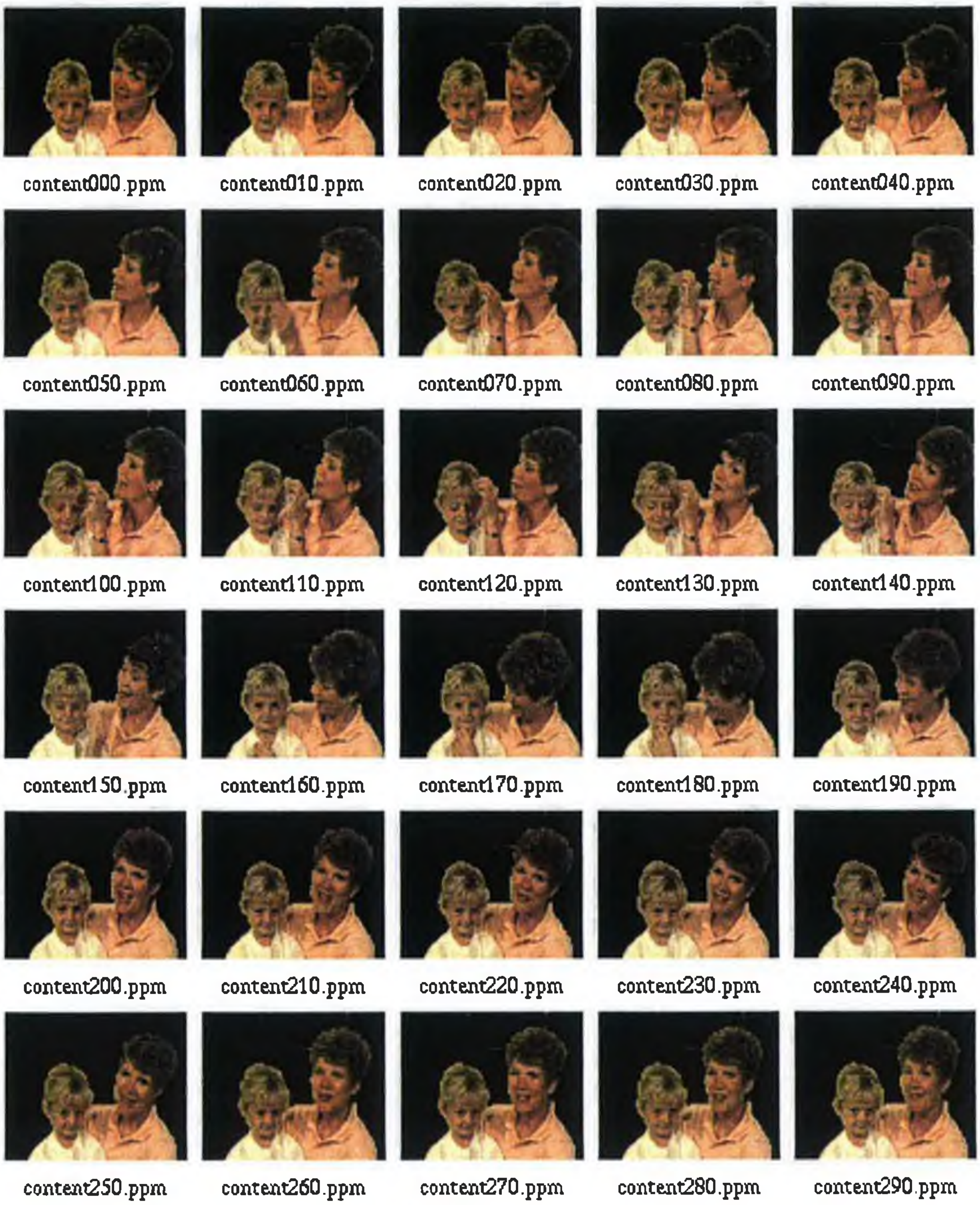


Fig 6.26 : Content of Interest determined as everything but background clusters



content000.ppm content010.ppm content020.ppm content030.ppm content040.ppm

content050.ppm content060.ppm content070.ppm content080.ppm content090.ppm

content100.ppm content110.ppm content120.ppm content130.ppm content140.ppm

content150.ppm content160.ppm content170.ppm content180.ppm content190.ppm

content200.ppm content210.ppm content220.ppm content230.ppm content240.ppm

content250.ppm content260.ppm content270.ppm content280.ppm content290.ppm

Figure 6.27 : 'Mother & Daughter' Content of Interest tracking result (~10 seconds of video)

This result shows good content of interest tracking even when new regions (i.e. mother's hand) enter the scene. Background clusters are relatively stable and sufficiently distant in Euclidean space to the foreground.

6.4.3 Salesman Result

Salesman is a much more challenging sequence to extract useful regions from using clustering. It also has periods of fast body movement making the tracking process difficult. For this test, foreground region clusters, described in table 6.2, were tracked. Both clustering and tracking were once again performed in Euclidean space but this time with no scaling of the *x,y* co-ordinate data. Acceptable T_{COI-1} and T_{COI-2} thresholds were found to be 0.185 and 0.1 respectively with cluster merge threshold set to 0.11. Figure 6.28 shows the content of interest as determined at the first frame of the sequence.

Table 6.2 : Foreground cluster descriptions and feature vector means for Salesman sequence (frame 0)

Cluster	Content	{y, u, v, x-coord, y-coord} region means
13	Hair	{0.106065, 0.715863, 0.558525, 0.602995, 0.190069}
20	most of face	{0.546135, 0.591626, 0.730949, 0.588473, 0.294895}
26	lower faces of box	{0.346857, 0.688007, 0.521821, 0.308902, 0.436146}
27	top face of box	{0.568199, 0.713017, 0.482539, 0.322739, 0.400712}
28	rest of face	{0.341763, 0.644232, 0.654221, 0.543730, 0.350819}
29	part of shirt	{0.652494, 0.888803, 0.388575, 0.649615, 0.560581}
34	part of shirt	{0.541976, 0.824156, 0.423699, 0.266114, 0.655029}
35	part of shirt	{0.711938, 0.880608, 0.392671, 0.403712, 0.548118}
36	part of shirt	{0.553364, 0.857580, 0.426913, 0.490500, 0.507115}
37	part of shirt	{0.796367, 0.898392, 0.374563, 0.670957, 0.483206}
38	part of shirt	{0.259927, 0.871875, 0.413943, 0.703214, 0.618444}
42	part of shirt	{0.325734, 0.808598, 0.432349, 0.326671, 0.680252}
43	left hand	{0.515911, 0.619891, 0.651962, 0.485408, 0.701861}
44	part of shirt	{0.462219, 0.863705, 0.405263, 0.690170, 0.686780}
45	part of shirt	{0.599980, 0.869030, 0.389523, 0.606809, 0.757575}
46	part of shirt	{0.735320, 0.877791, 0.384117, 0.768957, 0.686783}
50	Tie	{0.205380, 0.715000, 0.594819, 0.525122, 0.661739}

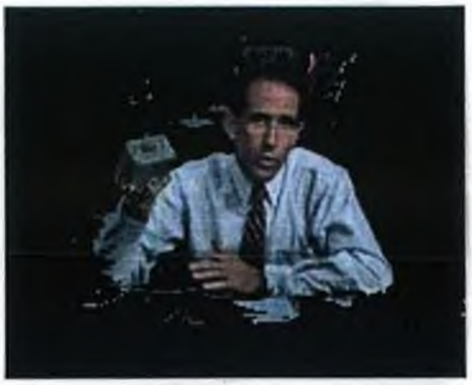


Figure 6.28 : Content-of-Interest determined as foreground cluster set listed above.

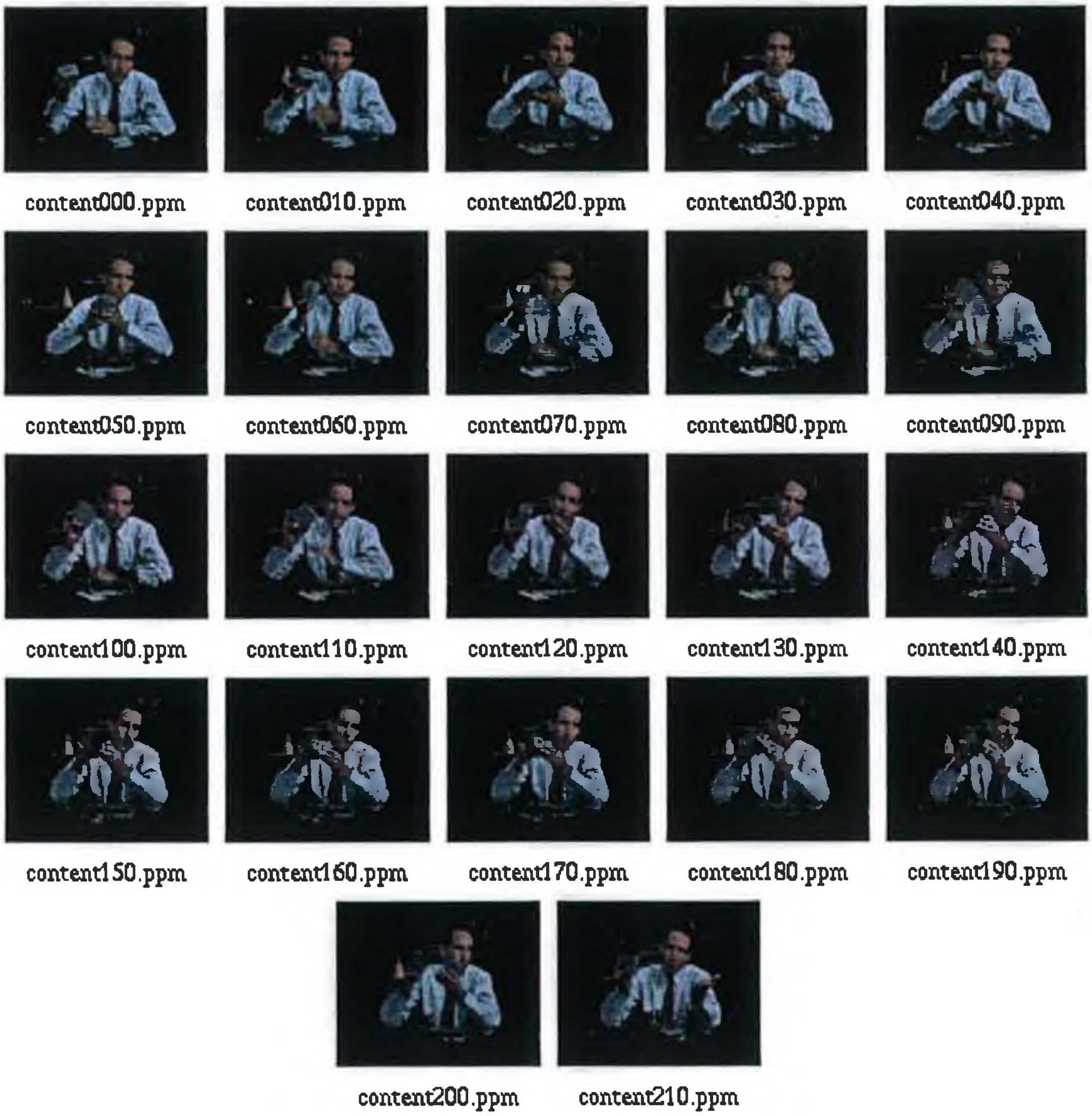


Figure 6.29 : 'Salesman' Content-of-Interest tracking result (~7 seconds of video)

The salesman result of figure 6.29 occasionally loses the hand regions and generally includes the unwanted reflection of the person in the table but this is still a useful result and it is achieved for a challenging sequence.

6.4.4 Foreman Result

The foreman sequence is a very active sequence with a lot of foreground and background motion. The best result here was obtained by tracking the background clusters and with the Mahalanobis distance clustering and tracking. Background descriptions and cluster vector means are shown in table 6.3. The cluster merge threshold here was the default 0.6 and an acceptable value for both T_{COI-1} and T_{COI-2} was found to be 24 in this space.

Table 6.3 : Background cluster descriptions and feature vector means for 'Foreman' (frame 0)

Cluster	Content	{y,u,v,x-coord, y-coord} region means
0	Left wall	{201.658234, 141.455704, 92.341774, 11.962026, 1.632911}
1	"	{212.026947, 117.312057, 132.212769, 27.954611, 11.697872}
2	"	{147.667740, 124.990326, 128.074188, 26.312902, 25.519356}
3	"	{215.029602, 121.359406, 131.266388, 73.109940, 10.902748}
4	Right wall	{123.234955, 117.266472, 132.320923, 131.850998, 14.724928}
5	"	{200.769653, 116.672989, 132.928635, 140.645889, 11.490515}
6	"	{138.243317, 119.632050, 132.789322, 153.359055, 32.385757}
7	Left wall	{142.091553, 116.086266, 132.720078, 30.494719, 27.029930}
8	"	{217.005814, 120.982536, 130.608841, 28.308498, 28.834692}
10	Right wall	{203.994003, 119.724136, 132.407791, 157.643173, 30.317841}
11	"	{196.873657, 116.705566, 133.078156, 138.905777, 42.871521}
13	Left wall	{215.904495, 121.930275, 129.880615, 25.588348, 58.138493}
17	"	{40.092438, 125.521011, 128.920166, 15.802521, 77.802521}
20	Right wall	{106.693291, 117.857635, 132.652771, 139.155090, 86.975693}
21	"	{202.818634, 118.838234, 131.549026, 148.502457, 68.770424}
22	Left wall	{195.043533, 119.037315, 130.921646, 22.400497, 92.281097}
23	"	{104.545631, 118.675728, 132.352432, 40.636894, 82.719414}
26	"	{200.265335, 117.830666, 131.563995, 161.226669, 96.830666}
27	"	{202.196518, 121.302666, 131.014694, 23.423517, 123.844856}
28	"	{129.729904, 122.855667, 129.286591, 44.076290, 113.164948}
31	"	{87.290871, 115.492004, 134.358948, 145.822586, 122.054665}
36	Part of cranc	{154.864868, 95.432434, 152.513519, 164.608109, 141.486481}

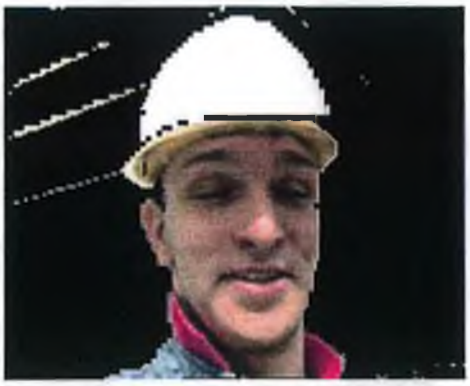
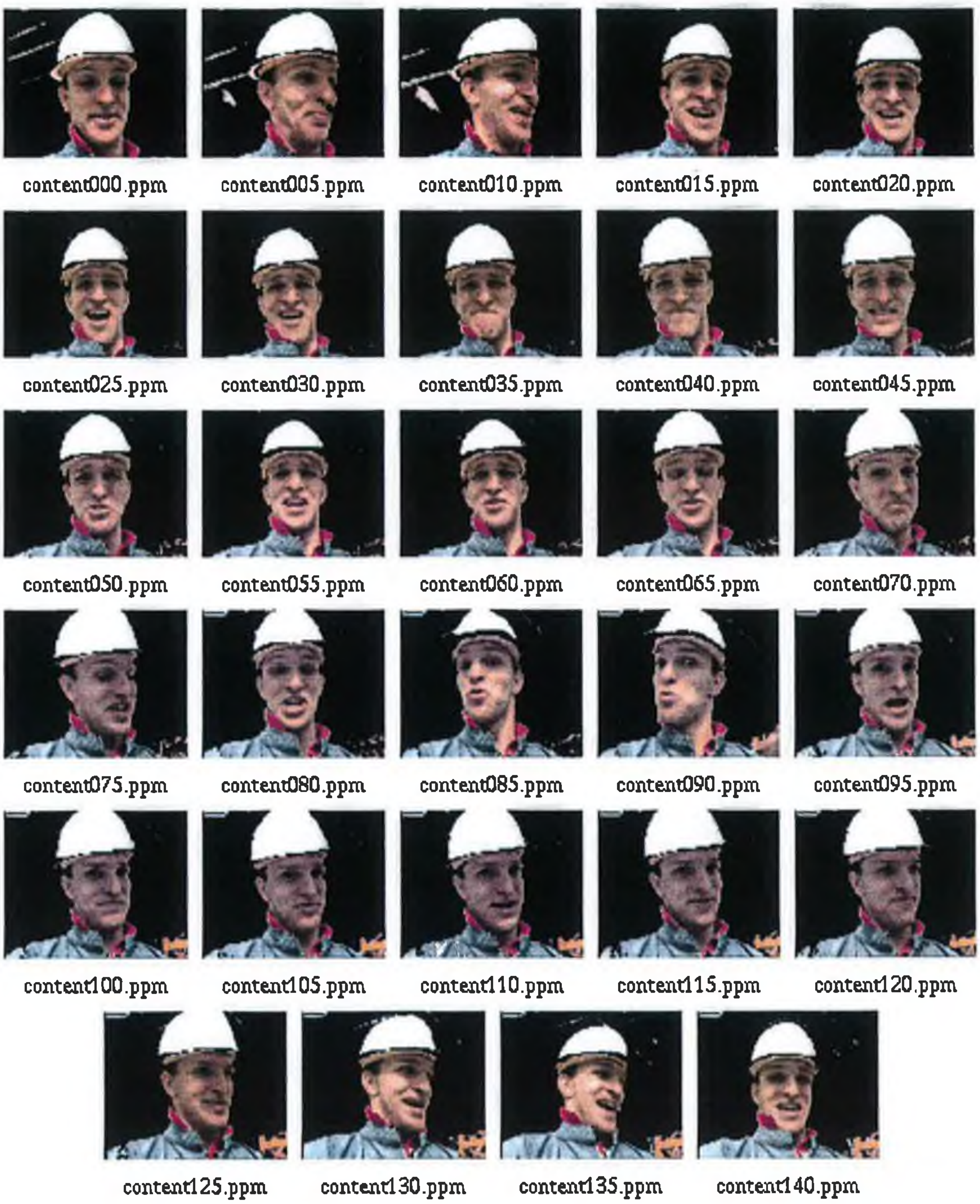


Figure 6.30 : Content of Interest determined as everything but background clusters



content000.ppm content005.ppm content010.ppm content015.ppm content020.ppm

content025.ppm content030.ppm content035.ppm content040.ppm content045.ppm

content050.ppm content055.ppm content060.ppm content065.ppm content070.ppm

content075.ppm content080.ppm content085.ppm content090.ppm content095.ppm

content100.ppm content105.ppm content110.ppm content115.ppm content120.ppm

content125.ppm content130.ppm content135.ppm content140.ppm

Figure 6.31 : 'Foreman' Content of Interest tracking result (~6 seconds of video)

The activity of the content-of-interest, the foreman, is evident in the result of figure 6.31, however, the tracking mechanism succeeds in extracting all but the brim of his hat from the building site background. Parts of the crane are also occasionally misclassified, but this is considered a generally good extraction result.

6.5 Clustering Review

Clustering of pixel feature vectors is a relatively straightforward technique where pixel property vectors are iteratively reassigned to cluster bins based on a distance metric between individual vectors and cluster means. The aim of this research work was to outline a simple but flexible clustering strategy that could be applied to image feature clustering and segmentation for content definition and tracking purpose. This has been achieved quite successfully with simpler image feature vectors than originally thought: 3 component colour and 2 component position information only.

Extension of this procedure to higher dimension feature vectors would not be a problem. Texture information in the form of Gabor filter responses [52], and depth information from stereoscopic imagery or structure-from-motion [53] work are readily available candidates. For cluster tracking stages in subsequent frames closed region size and principal components of the shape radii as described by [56] and [57] could also prove useful.

The clustering process is, admittedly, a heuristic one relying on a certain amount of human supervision, at least on initial sequence frames, to select the best initial conditions: distance metric, feature component scaling factor and cluster merge thresholds. For single frame segmentation by colour clustering, Pauwels et al. [58] proposes a fully unsupervised algorithm which makes no assumption about the underlying data distribution. Their work uses a particular convolution kernel, a difference of Gaussian kernel, allowing for better discrimination between clusters. The width of this kernel is estimated from the data using partial sample cross validation. Interestingly this kernel has a positive centre and a negative surround, a similar characteristic structure of the centre-surround receptive fields found in retinal ganglion cells.

The clustering segmentation work shows promise for the application of object-oriented coding, as envisaged by the MPEG-4 coding community. However it remains a bottom up segmentation process relying on a human observer to classify resulting regions. For semantic extraction and automatic characterisation of video sequence content it is necessary to combine this approach with a model or expectation based process. This also forms part of the motivation of the COST211^{quat} [20] simulation subgroup with the development of the Analysis Model described in the next chapter. The next chapter also brings the content definition and extraction research up to date with the latest developments in content analysis within the MPEG-4 and COST 211 groups and others.

7. Latest Content Definition Results

7.1 Introduction

This chapter brings the content definition and tracking results up to date with the latest developments in the COST 211 research and ISO/IEC MPEG standardisation groups. COST 211, introduced in chapter 2, is a European collaborative research group which has been active throughout its history with video coding, most prominently with the development of the ITU-T video coding recommendations H.261 and H.263, and the first collaborative efforts at content based coding with SIMOC-1. The latest efforts of the group, within the new COST 211^{quat} project, are a shift away from this coding heritage and focused much more on analysis of generic video specifically to generate input for MPEG-4 content based coders and MPEG-7 content classification and database query engines. The author's contributions to the COST 211$^{\mathrm{quat}}$ group were centred on the development of a collaborative test model, known as the *Analysis Model (AM)* [60, 61]. This AM, in the form of jointly developed and distributed software, is intended to provide the researchers within the group a common framework to compare and optimise visual analysis algorithms in an experimental approach. The AM results are also expected to serve as a benchmark to use when assessing alternative algorithms and technology.

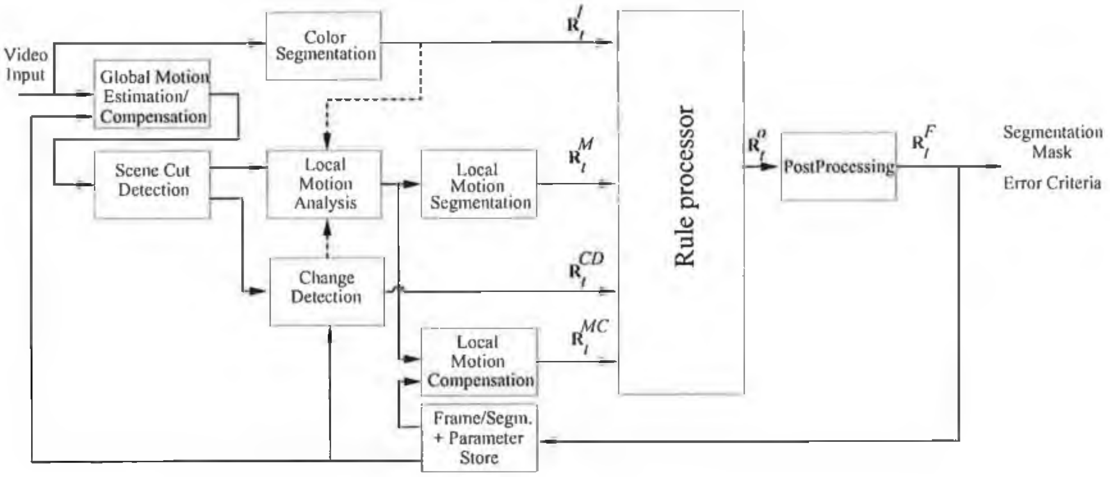
Within the latest (July 1998) version of the AM the *Recursive Shortest Spanning Tree (RSST)* algorithm [62] is used for both the colour and motion segmentation modules. Software for the colour RSST module was written by the author to include in the AM source distribution. Colour RSST segmentation and several alternative data driven and semi-automatic segmentation algorithms, including morphological watershed, pyramidal region growing and the feature clustering approach of chapter 6 have also been compared by the author in [63] as part of the COST activity. The AM and these other alternative content definition algorithms discussed within COST are further detailed in the following section.

The MPEG-4 group is concerned primarily with standardisation of the audio-visual object bit-stream and decoder specifications. The usual MPEG procedure is, however, to provide encoder and content production guidelines as an informational annex to the standard. Content definition through so-called video object plane generation tools are described in this annex and latest results of these techniques are also detailed here in section 7.3.

7.2 COST 211^{quat} Activities

7.2.1 COST Analysis Model

This framework fuses results from a variety of sources including motion, colour, and intensity change information with a rule based processor to determine an optimal segmentation result. Figure 7.1 below shows the model architecture. The following sections detail the individual components of this architecture.



R_t^I etc. are image size label maps, binary and multi-valued.

Figure 7.1: Model Architecture from the COST211^{quat} Analysis Model (v. 4.0) [60]

7.2.1.1 Global Motion Estimation / Compensation Module

This first module estimates apparent camera motion between successive input frames and compensates for this before further processing. Estimation is carried out by fitting an eight parameter affine motion model to an assumed rigid planar object. Where this assumption is invalid the estimation step will fail. Only background pixels are used in a regression estimation technique and a further assumption is made on initial or scene cut frames as to the position of background pels. Camera motion compensation is carried out using bilinear interpolation, but this is only carried out if a moving camera is judged to be the most likely source of global motion. A heuristic is defined in the AM to make this decision.

7.2.1.2 Scene Cut Detection

Motion compensation is not useful in the case of a scene cut between two consecutive frames. A very simple scene cut detection module is used here which evaluates the difference (mean square error) between the current frame and the camera motion compensated previous frame. If this difference exceeds a defined threshold the segmentation algorithm is reset.

7.2.1.3 Change Detection

The change detection module, as with SIMOC-1 (Chapter 2), defines an image mask distinguishing between changed and unchanged image regions. In this module, also described in [64], an initial change detection mask is computed by a threshold operation on the squared luminance difference of the current frame and the previous camera motion compensated image. To calculate the threshold to use, a significance test is carried out with pixel differences assumed to correspond to camera noise (Gaussian) and not to moving objects. For a luminance difference d_i assumed to correspond to Gaussian distributed camera noise (null hypothesis H_0) the probability distribution is given in eq. 7.1 with the variance σ^2 being twice the variance of the camera noise.

$$\mathrm{p}(\mathrm{d}_i|H_0) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{d_i^2}{2\sigma^2}\right) \qquad \text{(Eq. 7.1)}$$

All pixels for which T (defined in eq. 7.2 as the variance normalised sum of pixels in the squared difference image within an n x n window) exceeds a threshold t_α, are marked as changed. The others are marked as unchanged.

$$\mathrm{T} = \frac{1}{\sigma^2}\sum_{i=1}^{n\cdot n} d_i^2 \qquad \text{(Eq. 7.2)}$$

An iterative relaxation of the initial change detection mask is carried out to help enforce spatial homogeneity on the moving object boundaries. A maximum *a posteriori* detector is used in each iteration of this relaxation process to determine if boundary changed pixels are due to moving objects or not. The probabilities of individual boundary pixels belonging to changed and unchanged regions are again modelled as Guassian distributions and both *a priori* probabilities are modelled by a

Markov random field (MRF) of the local 3x3 pixel neighbourhood. Horizontal / vertical neighbours and diagonal neighbours are assigned different potentials in this MRF.

The final step within the change detection module is designed to promote temporally stable object regions. This is done by building up a frame memory store of all changed pixels over the last L frames, where L denotes the memory depth. If a pixel in the previous object mask, R_0, was marked as changed and it was also marked as changed in one of the last L frames before R_0, then the same pixel in the current object mask is also marked changed. The value of L is linked to the amount of motion of the moving objects in the scene and the size of those objects. A heuristic is defined in the AM to change this depth dynamically with changing displacement vector amplitudes and moving object sizes.

The predefined parameter set for the change detection mask generation module is considerable and includes the local window squared difference threshold t_α, the MRF potentials, the maximum number of iterations in the relaxation step and the frame memory store depth. The heuristics also change for QCIF and CIF sequences.

7.2.1.4 Adaptive Frame Skip and Interpolation

Insufficient, or excessive, motion between frames will have an impact on the performance of the motion analysis and rule-based stages of the AM. The function of this module is to choose a frame pair exhibiting just sufficient motion to allow some useful motion segmentation data to be obtained. To solve this problem the amount of motion between successive frames at full frame rate i.e. frame$_n$, frame$_{n+1}$, frame$_{n+2}$ etc. is measured and only the first frame pair exceeding a defined threshold is propagated to the rest of the model. Object masks are requested at a defined frame rate, however, so masks for any skipped frames must be interpolated from the neighbouring masks. This is done in a similar manner to MPEG-2 B-frame interpolation using the motion vectors for each pixel and the scaled contributions of the neighbouring object masks.

7.2.1.5 Colour Segmentation

The colour segmentation step applies an image partition algorithm to the current input image frame to segment into semantic content regions. Any image partition algorithm could be used in this module. However, the latest AM definition specifies the recursive shortest spanning tree algorithm [62]. RSST allows simple control over the number of regions and resulting segmented detail. Section 7.3 details this method and comparisons with several others including the author's feature clustering approach.

7.2.1.6 Local Motion Analysis

Two motion estimation methods, with differing computational requirements, are specified by the AM. The first is the hierarchical block match algorithm described by Bierling [18] and detailed in section 3.3.1. The second uses a Gibbs formulation described in detail in [65]. This algorithm essentially minimises an energy function built using an estimated motion vector field and colour segmentation information for each frame.

7.2.1.7 Local Motion Segmentation

Two models are also specified for the motion segmentation module. The first uses the same RSST algorithm as the colour segmentation module, see section 7.3, but here, the distance measure is defined as:

$$d(R_1,R_2) = \left\| \mu_{R_1} - \mu_{R_2} \right\|^2 \frac{N_{R_1} \times N_{R_2}}{N_{R_1} + N_{R_2}}, \mu = \begin{bmatrix} M_{x,avg} \\ M_{y,avg} \end{bmatrix} \quad \text{(Eq. 7.3)}$$

Here, μ represents in this case the average translational motion vectors over a region R and N represents the region size.

The second model assumes an affine (6-parameter) motion model for the motion vector field and attempts to extract regions with a successful affine parameter fitness level. This model accommodates zoom and rotation as well as translational motion and generally performs better catering for complex real-world object motion.

7.2.1.8 Local Motion Compensation

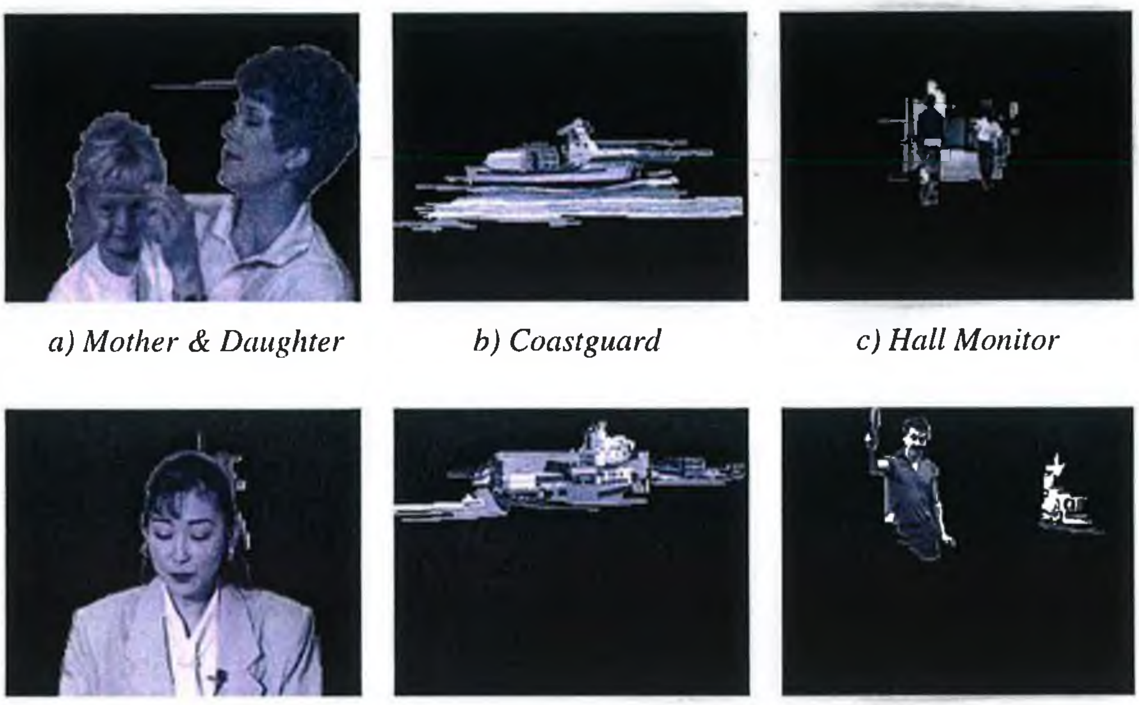
The motion compensation module gives the AM tracking capability and supplies further object semantic information (e.g. object is halted or newly exposed) to the rule processor. This module uses the previously determined segmentation result and the currently estimated motion information to generate a temporal prediction of the current segmentation mask. The result of this stage is $\mathbf{R}_t^{MC}$ in figure 7.1.

7.2.1.9 Rule Processor

The Rule Processor module fuses the segmentation results of the colour segmentation, $\mathbf{R}_t^{I}$, and motion segmentation, $\mathbf{R}_t^{M}$, modules as well as the change detection mask, $\mathbf{R}_t^{CD}$, and the estimated object mask (motion compensated previous result), $\mathbf{R}_t^{MC}$. Each of these inputs supplies different but critical information to the rule processor. $\mathbf{R}_t^{I}$ will typically contain over-segmented but accurate content boundaries. $\mathbf{R}_t^{M}$ should distinguish between disjoint moving objects, $\mathbf{R}_t^{CD}$ provides a generally reliable foreground / background estimate and $\mathbf{R}_t^{MC}$ provides tracked object information.

The rules to combine all this information currently fall into two categories or operation modes. The first operation mode is a foreground / background classification scheme relying mainly on the change detection input but also using the local motion analysis information and the colour segmentation information to determine the final binary mask output. The second operation mode attempts to go further by distinguishing between objects with different motion. It does this by applying rules to determine individual tracked objects and newly exposed objects but also reconfirm existing objects are still coherent in motion. The multi-level segmentation mask produced as an output of this second operation mode is further processed to reclassify very small regions, remerge neighbouring and coherently moving regions and apply a morphological filtering element to refine object edges.

Figure 7.2 shows some illustrative results of version 1.0 of the Analysis Model from [66]. The *Mother & Daughter* and *Akiyo* content objects are completely extracted in these example frames with only small misclassified regions of the background included. The *Coastguard*, *Hall Monitor* and *Table Tennis* results also show complete object extraction, however, this time a higher proportion of the background is included. The *Container Ship* result is the only example where parts of the foreground object are actually missed in the extraction. It also includes some areas of the unwanted background.



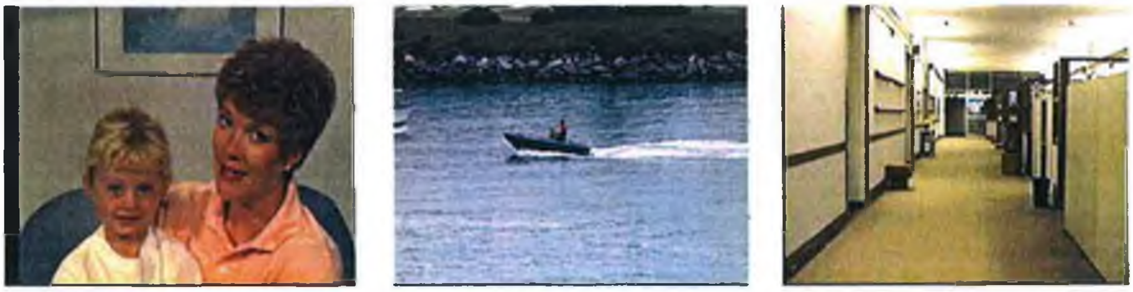
a) Mother & Daughter *b) Coastguard* *c) Hall Monitor*

d) Akiyo *e) Container Ship* *f) Table Tennis*

Fig 7.2 : Sample content-of-interest segmentation results of AM, from [66].

7.2.2 Semi-Automatic Definition Algorithms

Several semi-automatic definition algorithms, including the RSST algorithm within the AM, were investigated by the author in [63] and are further detailed here. An overview of each method and its definition performance results on the test set of figure 7.3 are provided.



a) Mother & Daughter *b) Coastguard* *c) Hall Monitor.*

Fig 7.3 - Original QCIF first frames

The test sequences used originate in the ISO/IEC MPEG and ITU-T standardisation bodies. QCIF resolution sequences were used with input colour format of 4:2:0. Human intervention was used to specify application specific target and other parameters for these frames. The goal for each algorithm was that all important content edges should be defined.

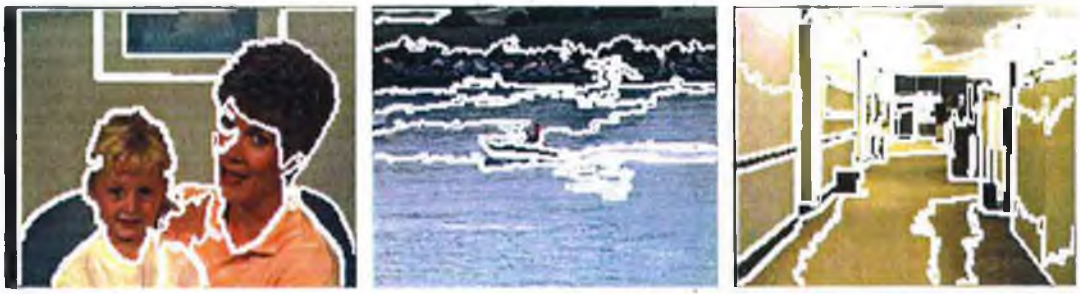
7.2.2.1 Recursive Shortest Spanning Tree

The RSST technique works by recursively grouping neighbouring regions based on a defined link weight. In the colour segmentation case this link weight is the squared colour vector (*YUV*) difference with a scale factor biased against the merging of large regions. Specifically for two adjacent regions, R_l and R_2, the distance between them is given by equation 7.4.

$$d(R_l,R_2) = \left\| \mu_{R_1} - \mu_{R_2} \right\|^2 \frac{N_{R_1} \times N_{R_2}}{N_{R_1} + N_{R_2}}, \mu = \begin{bmatrix} Y_{avg} \\ U_{avg} \\ V_{avg} \end{bmatrix} \qquad \text{(Eq. 7.4)}$$

where μ is the feature vector representing the mean *Y*, *U* and *V* values of all pixels inside the region and *N* is the number of pixels within the region. Figure 7.4 shows the boundary images obtained using RSST on the COST211^{quat} test data set. As mentioned in section 7.2.1.7 the technique can also be applied to motion vector fields using motion feature vectors. The algorithm steps are set out here.

```
• Initialisation
        No of Regions = Total number of pixels in image.
        Size of all = 1
• Loop While (No of Regions > User Specified Target)
        Calculate distance measure for all 4-connected region pairs
        Join two closest regions
        Calculate new region values and sizes
        Decrement No of Regions
```



a) target = 14 *b) target = 9* *c) target = 45*

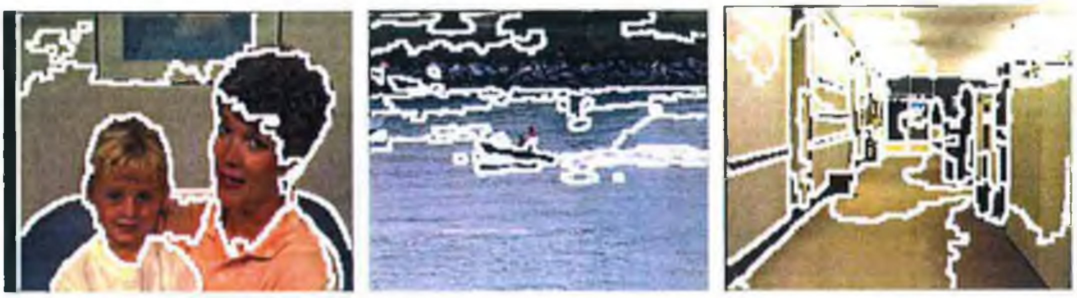
Fig 7.4 - RSST Test Set Results with variable target number

Although some arbitrary region boundaries are shown, this is an encouraging result, with most important content boundaries detected. Algorithm speed on a reference system (Solaris 2.5, Sun SPARC Station 20 platform) was an average of 3.2 seconds per frame.

7.2.2.2 Pyramidal Region Growing

This technique is an extension of the scheme, proposed in [67], which uses a hierarchy of frames of increasing resolution and a competitive region growing process to produce a final segmentation. The basic method involves forming a truncated image pyramid, with each layer having a quarter of the nodes of the layer below. Each node has both a bottom-up (BU) and a top-down (TD) parameter associated with it and these are assigned in an iterative process. The BU parameter is an image measure (e.g. mean grey level) related to the associated 4x4 'son' nodes of the next larger layer. The TD parameter at the top layer is initially the BU parameter of the same node. For the next and subsequent layers the TD parameters are forced to take on one of the TD parameters of the nearest 2x2 'father' nodes on the upper (smaller) layer. The one with the closest BU parameter as the node under question is chosen. At the bottom layer the TD and BU values are made equal forming a partially segmented image and the process is repeated until stable (i.e. TD and BU parameters equal throughout pyramid). Where the pyramid is truncated determines the scale of the features extracted.

A number of problems with this basic technique have been addressed by Beaumont in [68] such as the poor choice of region values, failure to segment 'slender' shaped regions and failure to concurrently segment regions of varying size. A mechanism to seed and erode new regions at different pyramid layers and an edge detection algorithm, to prevent arbitrary segmentation in areas of low contrast, was incorporated by Beaumont and his implementation was used in these tests. The parameters, SEED and ERODE, were adjusted for the best segmentation of the first frame of the sequence. The SEED threshold controls the number of new regions being seeded at each resolution whereas the ERODE threshold controls the amount of edge strength required to establish a new region. First frame results are given in figure 7.5.



a) S/E : 3/5 *b) S/E : 1.1/5* *c) S/E : 1.1/5*

Fig 7.5 - Pyramid region growing result, variable seed/erode parameters

This technique performs well considering that processing is only carried out on the luminance component of the test sequences. Algorithm took an average of 13.5 second/frame on the reference platform.

7.2.2.3 Morphological Watershed

Mathematical morphology is a well developed technique investigating the properties of size, shape and structure of mathematical set objects known as *complete lattices.* A lattice is a mathematical set with an associated ordering relation and the definition of a complete lattice is given by Czerepiński and Bull [69] as one where every subset also has an infimum and a supremum. Two lattices are considered useful in the context of image processing, the set of all subsets of the Euclidean plane, used in binary morphology, and the set of all functions on the Euclidean plane, used in the greyscale image morphology considered here. The erosion, dilation and the morphological opening and closing operations are also defined in [69] for any surface function within a defined neighbourhood window.

The following analogy is often used in the description of the watershed morphological approach where a greyscale picture is considered to be a physical topographic surface with a pixel's grey level representing its elevation at that point. The watershed segmentation process locates the regional minima of the surface, pierces holes at these points and then slowly immerses the surface into water. The different catchment basins of this surface are progressively filled during the immersion, starting from the minima of lowest altitude. Where the water coming from two different basins would merge, if immersion continued, a "dam" is "constructed" before further immersion and this is repeated until the surface is completely immersed. All the regional minimas are then surrounded by these "dams", which mark out the catchment basins, or regions, of the picture. A fast implementation of the watershed algorithm, based on [70], was used in this comparison. In this implementation, before the watershed algorithm is applied, several morphological filtering operations are performed to simplify the input image. These initial steps are:

- Filtering of the original image with the morphological open-close by reconstruction operator with a specified filter size.
- Calculation of the gradient of the filtered original by application of erosion then dilation operations using a 3x3 structuring window and calculating difference.
- Labelling of the flat regions of the gradient in a 'marker' image.
- Preservation of the highest peaks only in the gradient between 'marker' regions

It is on this modified gradient that the core watershed segmentation is finally applied. To ensure that as many as possible of the required content edges were preserved, a low morphological filter size of 5 was chosen for the test set. Results are shown in figure 7.6.

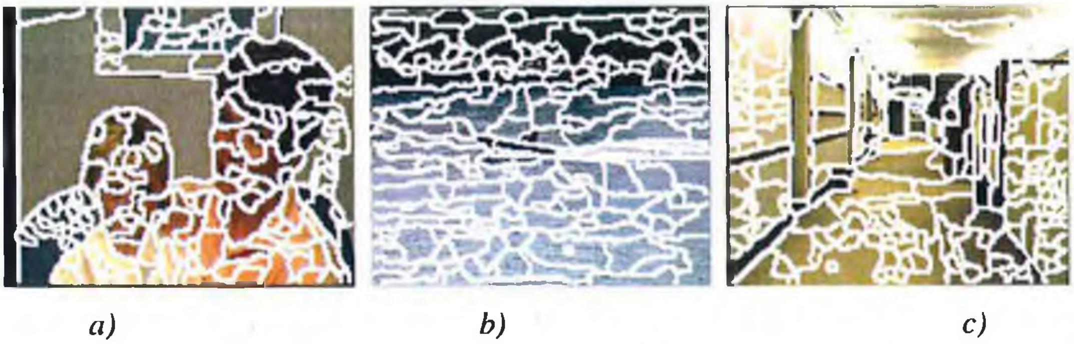


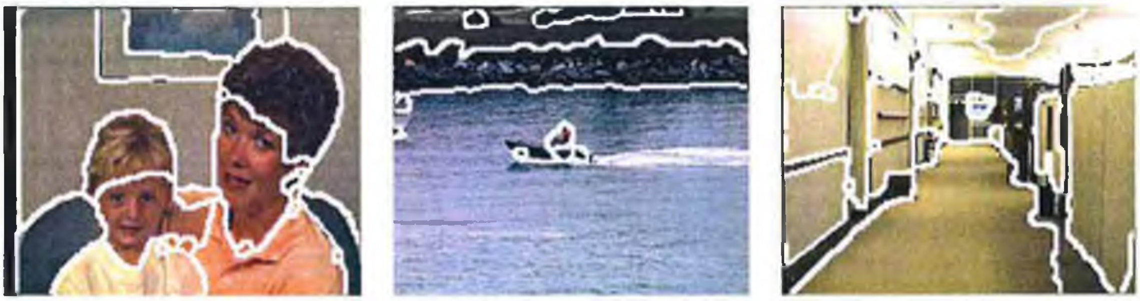
Fig 7.6 - Morphological watershed, filter size of 5

The morphological watershed implementation was fast with an average of 5.1 second/frame on the reference platform. The results show many arbitrary boundaries but this is expected with such a small morphological filter size. At higher filter sizes there are fewer regions but also important content edges are removed. The arbitrary boundary regions could be merged in further processing, such as RSST, while retaining the good boundary edges.

7.2.2.4 Feature Clustering

The feature clustering algorithm is detailed in Chapter 6. The input to the clustering process consists of a data set of pixel property feature vectors and the technique iteratively re-assigns these vectors to distinct cluster bins based on a distance metric between individual vectors and the cluster means. The cluster label images are post-processed, in the results presented below, with the fine watershed of the previous section.

Clustering based on a distance metric is very close to the RSST approach except here there is no explicit connectivity constraint. For this test set, feature vectors with both colour and positional data were used. Both Mahalanobis and Euclidean distance metrics and the 8 and 64 cluster initialisation were also tried to obtain an optimal result.



a) K=8, T=0.11, Maha. {y,u,v,x,y} custering

b) K=8, T=0.11, S=0.4, Eucl. {y,u,v,x,y} custering

c) K=64, T=0.2, S=1.0, Eucl. {y,u,v,x,y} custering

Fig 7.7 - Feature clustering and watershed post-processing

Fig 7.7 a) was obtained using an 8 stripe initialisation, intensity, colour and position {y, u, v, x, y} feature vectors and Mahalanobis clustering with a threshold of 0.11. The only important content boundary missing in this case is the right side of the child's face. Otherwise this is a generally useful result. Fig 7.7 b) was obtained also using an 8 stripe initialisation but this time using Euclidean clustering with positional information scaled to 40% and a threshold of 0.11. The water and background is well defined here but the upper boundaries of the boat are not retained. For fig 7.7 c) position information was not scaled but Euclidean clustering was used in addition to a 64 block initialisation. Result is acceptable with most floor and wall boundaries defined.

7.2.2.5 Comparisons

The goal of this processing is to delineate true content edges while giving as few arbitrary boundaries as possible. The RSST technique performs well on the generic video test set with all content edges defined but also some arbitrary edges for this test set. The feature clustering with watershed post-processing and the pyramid region growing technique both also perform well and have fewer arbitrary boundaries than RSST but there is also a corresponding reduction in real boundaries. In terms of complexity the RSST is the fastest algorithm with the watershed next in speed.

Arbitrary boundaries are a problem particularly for the watershed algorithm. Region merging algorithms based on graph theory could also be applied here as a post-process but this was not investigated. Arbitrary boundaries are only specifically addressed in the pyramid region growing approach where the gradient through a region boundary is computed. This could also be investigated as a general post-process for all these content definition techniques.

7.3 MPEG-4 Content Generation Guidelines

One of the main distinctions of the MPEG-4 video coding framework is the facility for content-based functionalities. These in turn require the description of the visual scene in terms of Video Objects (VOs) or sequences of Video Object Planes (VOPs). MPEG-4 encoding engines wishing to exploit this flexibility for generic frame based video will require local segmentation techniques to automatically or semi-automatically identify the objects appearing in the sequence. Automatic VOP generation techniques will be required for real-time applications but assisted segmentation may also be useful for offline content production.

It is only MPEG-4 decoders which must be fully specified by the standard itself but several automatic and semi-automatic segmentation schemes will be provided as an informational annex to assist content generators. The details of the following sections is drawn from [72], a proposal by 3 companies (Fondazione Ugo Bordoni, University of Hannover, and the Electronics and Telecommunications Research Institute of Korea), describing several techniques useful for both automatic and semi-automatic segmentation. One of these proponents – University of Hannover, is also active within the COST 211 AM definition and similar elements can be seen in the relevant description.

7.3.1 Temporal segmentation using higher order moments and motion tracking

This algorithm processes a group of frames to derive the segmentation mask for each one. Pre-processing for each frame is only carried out to remove global motion components. Thereafter the algorithm applies the three separate steps shown in figure 7.11 to derive a video object mask for that frame.

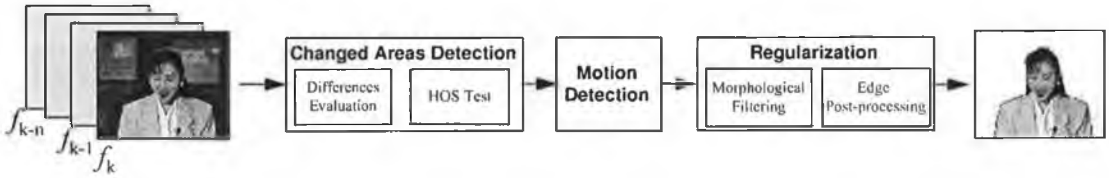


Figure 7.11: Temporal segmentation using higher order moments and motion tracking

The algorithm breaks down to a changed areas detection module, a motion detector and a regularisation stage. The changed areas detector calculates for each pixel a fourth-order moment of each frame difference, *d(x,y)* within a 3x3 window. See equations 7.5 and 7.6. This is then compared to an estimate of background activity [73] in order to reject any luminance variations due to noise. The result of this stage is a thresholded fourth-order moment map, denoted as a Higher Order Statistics (HOS) map in this algorithm, on which the motion detection process is carried out.

$$\hat{m}_d^{(4)}(x,y) = \frac{1}{9} \sum_{(s,t)\in\eta(x,y)} (d(s,t) - \hat{m}_d)^4 \qquad \text{Eq. 7.5}$$

$$\hat{m}_d(x,y) = \frac{1}{9} \sum_{(s,t)\in\eta(x,y)} d(s,t) \qquad \text{Eq. 7.6}$$

Here, $\hat{m}_d(x,y)$ is the 1st order moment and *(s,t)* is an element of the 3x3 neighbour-hood of *(x,y)*.

The motion detection module function is to determine which changed areas result from uncovered background, which remain still, or which are moving objects. Up to three previous HOS maps are compared with a sum of absolute difference criteria applied within a 3x3 window. Null displacements on all HOS map pairs indicates this area is still and should be designated uncovered background.

The regularisation module finally reclassifies small still regions internal to moving objects and performs morphological filtering and other post-processing techniques to clean up the object mask.

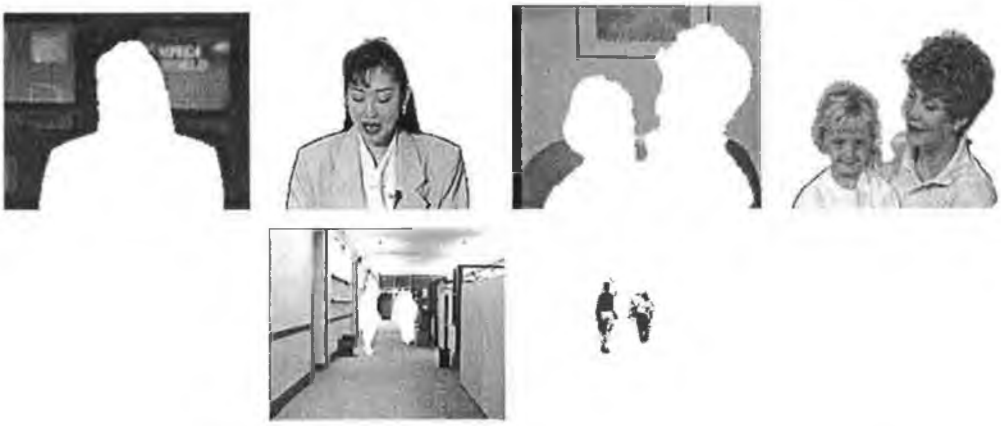


Figure 7.12: Fondazione Ugo Bordoni segmentation results [72] for akiyo, mother&daughter and hall-monitor

Results of the described method were shown on several meetings of the ISO/MPEG-4 Video Group. Some experimental results are shown above in fig. 7.12.

7.3.2 Temporal segmentation using change detection and luminance edge adaptation

This algorithm is the Hannover proposal with many elements in common with the COST AM modules already described. Figure 7.13 outlines the main stages.

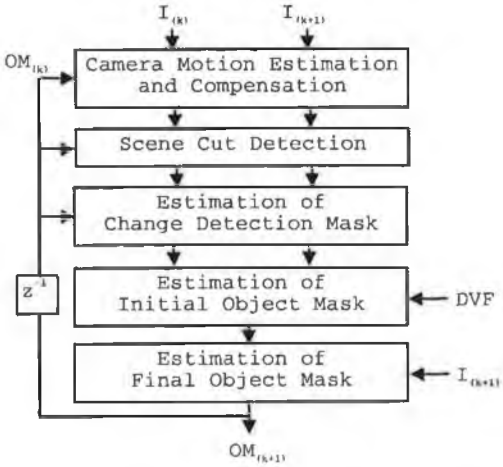

$I_{(k)}$
$I_{(k+1)}$
$OM_{(k)}$
Camera Motion Estimation and Compensation
Scene Cut Detection
Estimation of Change Detection Mask
z^{-1}
Estimation of Initial Object Mask
DVF
Estimation of Final Object Mask
$I_{(k+1)}$
$OM_{(k+1)}$

Figure 7.13 : Principle block diagram.

The camera motion estimation and compensation stage uses the same eight parameter model described earlier. The scene cut detector compares the mean square error of the current original frame and the camera motion compensated previous frame and if a threshold [74] is passed the algorithm parameters of subsequent stages are reset to initial values. The change detection modules consist of an initial change detection mask computation and a final estimation stage using the frame memory concept to define temporally stable object regions. Simplification of the mask and elimination of small regions are post processes applied before mask output. Some experimental results are shown in figure 7.14.

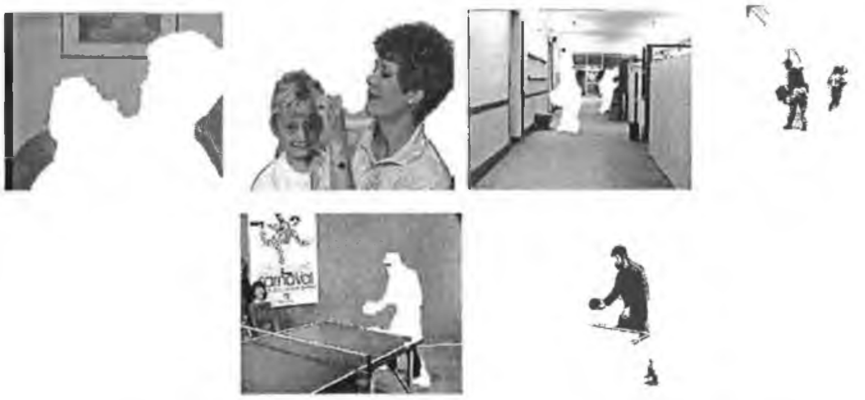


Figure 7.14 : Uni-Hannover segmentation results [72] for mother&daughter, hall-monitor and table-tennis

7.3.3 Spatio-temporal segmentation using morphological algorithm

The previous algorithms have very similar approaches. This algorithm, however, uses morphological tools and the watershed technique described in section 7.2.2.3 as the core region definition step. The main algorithmic steps are set out in figure 7.15 below.

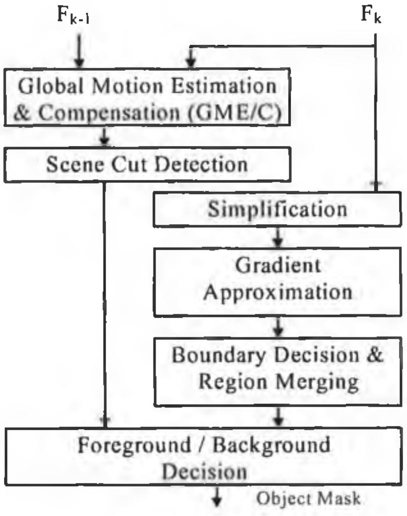

F_{k-1}
F_k
Global Motion Estimation & Compensation (GME/C)
Scene Cut Detection
Simplification
Gradient Approximation
Boundary Decision & Region Merging
Foreground / Background Decision
Object Mask

Figure 7.15 : Block diagram

The first stages mirror those of the previous algorithms. The main differences are a six-parameter affine model used for the global motion estimation and compensation stage and a mean absolute difference criteria for the scene cut detection. The image simplification stage applies morphological filters to remove regions smaller than a given size but preserve the contours of the remaining objects. The spatial gradient of the simplified image is then calculated and supplied as an input to the watershed process described earlier. Colour information is also incorporated into the gradient computation [75] to reduce ambiguous boundaries. The problem of over-segmentation of the watershed algorithm, seen earlier in section 7.2, is solved here by applying a region merging algorithm, also described in [75], based on graph theory and similar to the RSST approach [62]. The last stage of this process is the Foreground/Background decision module. This generates and overlays a change-detection mask on the current segmented regions. Where the majority of any segmented region is within this mask the whole region is denoted as Foreground.

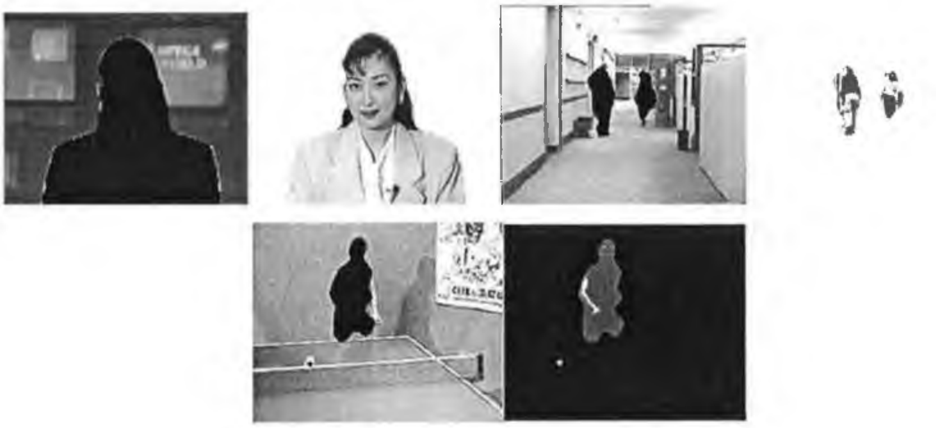


Figure 7.16 : Electronics and Telecommunications Research Institute of Korea segmentation results [72] for akiyo, hall-monitor and table-tennis.

7.4 Conclusions

Content definition has been, and continues to be, an active field of research. This is largely due to the MPEG-4 and MPEG-7 standardisation efforts and the new content focussed functionalities above and beyond simple compression. The COST 211 research group has changed focus specifically to address this new opportunity.

Complete content analysis frameworks capable of inferring content-of-interest from generic video whilst completely free of human intervention are the holy grail of this problem area and have proved as difficult to find. In many ways the difficulties here are imposed more by the nature of the original content and the fact that all analysis is restricted to work only with the information available. The human visual system has stereo vision, less spatial and temporal sampling problems and supporting acoustic information to work with. In contrast our analysis frameworks are completely deaf and have only one eye with tunnel vision. The problem statement is also ill defined. The content-of-interest is, by definition, a user dependant quality and will always depend on some degree of user interaction.

Frameworks that provide the most effective user assistance are the more realistic focus now and this is the area the COST Analysis Model and the various MPEG-4 generic VOP generation algorithms try to address. Several alternative frameworks are still being examined by the COST group including [71] and [76], however, the group is also collaborating on the development of one modular framework with parallel low level pixel analysis processors computing region segmentations before a final rule driven composite segmentation. The pixel processing modules themselves vary in complexity and definition performance but essentially perform the same function. That of simplifying the input data to a set of useful regions ready for later rule based combination.

The AM rule based approach has proved effective but the definition of the rule system here is critical. We risk forcing a pre-defined notion of content-of-interest if these rules are wrong. For example, the change detection module in the AM has a specific notion of one static background object and mostly singleton moving foreground objects. This was also the model for the SIMOC object based compression encoder and can work well for specific scenes but should not be imposed on other scenes for which it makes less sense. Even motion generally, whilst important, can not be the sole differentiator of content of interest. An accurate segmentation may still be required even when an object stops moving. Virtual avatar applications, as an example, representing real but segmented people and re-compositing in a shared environment, would not ideally want chunks of real texture suddenly disappear as soon as the real people sat down. This is addressed to some extent within the Change Detection AM module with the memory concept but it is more of a change in the rule system that is required emphasising, for the virtual avatar example, more texture and shape coherency than common motion.

Another argument that can be made, however, is that we can not clearly specify these rules as a predefined list or even apply them to a predefined hierarchy of partial segmentation results. The qualities that make something the content-of-interest at a certain moment may not be totally obvious even to the user. Appropriate interfaces, perhaps with smart user agents able to dynamically modify rules and weightings of attributes, will be a key component of any final analysis model framework.

The MPEG-4 standard itself, the spark and focus for a lot of this research and the raison d'être for the COST 211^{quat} Analysis Model, is due for imminent release as an International Standard in January 1999.

8. Conclusions

8.1 Overview

The lifetime of this research project has seen two dramatic developments in the area of digital video coding. The first has been the progress of compression research leading to a factor of two improvement over existing standards, much wider deployment possibilities and a new international ITU-T recommendation. The second has been a radical change in the approach to video content production and encoding with the introduction of the content based coding concept and the addition of scene composition information to the encoded bit-stream within yet another international standard. Smart and standardised decoders are now capable of manipulating actual arbitrarily shaped video content right up to the point of display. The new demands on content based video encoders and the analysis of generic frame based video for content segmentation has consequently been a hot research topic for a number of groups around the world. Contributions to both of these key developments and research areas have been made and reported on in this thesis.

8.1.1 Compression Results

A review of compression techniques in Chapter 2 identified both the standardised coding techniques and the most promising high compression proposals from a number of groups around the world. The maturity and generic nature of the standardised techniques were their main strengths. They were the fruits of a number of years of research into statistical and perceptual redundancy reduction and were even then implemented in a number of successful and practical multimedia services. Their main weakness was that, due to low complexity and real-time operation demands of many of these applications, the video model used by the standardised algorithms was necessarily a very simple one; using only low complexity translational only motion models and block residual coding regardless of picture content. Model-based coding, Layered coding and Object based coding approaches, on the other hand, were novel coding techniques utilising much more *a priori* knowledge about the scene content to push compression further. All these new approaches changed the generally accepted block-based video model to various more complex models. Evidence at that time suggested the more complex and object based video models, in particular those proposed by the Hannover group [8], did hold the promise of higher coding efficiency. This then led to the development within COST 211ter of the common SIMOC-1 [19] reference simulation model for object based coding. There was also evidence, however, that optimisation of standardised techniques could also be made and development of a new ITU-T Recommendation, then known as H.26p, was also started. Further compression research by the author centred around these two activities.

Chapter 3 looked at the development of the SIMOC model and extensions by the author particularly with regard to motion analysis and motion modelling. Motion played a much bigger role here than the standard ITU-T or ISO approaches where motion estimation and compensation is followed by a residual error coding stage. In SIMOC, if the distortion error was judged acceptable, motion parameters alone would be used to synthesise the decoded frame. Two areas were specifically looked at: motion vector accuracy with a proposal for a confidence measure of motion estimates and closer evaluation of the trade off in this codec between motion, shape and colour parameters. Motion estimation within SIMOC followed a similar approach to block based coders estimating a motion vector grid. The vector criteria here was the minimum mean absolute difference (MAD) between a block in the current frame and blocks within a search window in the last. Full block searching, within real-time codecs in particular, is too computationally expensive and sparse search motion estimation is typically employed. Accuracy then of sparse search algorithms is a problem. Two characteristics of block matching MAD error surfaces leading to poorly estimated vector fields were identified. One was the existence of local minima which could lead hierarchical search type motion estimators in the wrong direction at an early stage, and prevent location of the true minimum. The second was the common occurrence of valley shaped minimum regions, which leave the motion vector poorly defined along the direction of the valley. This relationship between image orientation and motion estimate accuracy is explored in Kearney et al. [89] and Nagel et al. [90]. The Hessian operator, a feature location tool, was found to be a useful confidence measure of estimated motion vector accuracy. This was combined with a general vector field smoothing technique from Stiller and shown to be effective in correcting erroneous vectors and influencing a smoother and hence lower entropy vector field. The combination of confidence measures, Markov modelled vector fields and higher density vector grids was found to be even more effective and achieved an optimal trade-off particularly between motion and colour parameter bitcounts. A marginal average ½ - 1dB PSNR improvement in decoded picture quality was demonstrated with these techniques over the original SIMOC specification with the added likelihood of further 1 – 2dB PSNR gains with better entropy coding of the motion parameters.

The motion estimation work reported on here was studied within the context of SIMOC but had general applicability to standard block based encoders. Work into optimisation of these and other standard block-based techniques for purely compression functionality was also underway in parallel to the object based coding developments. This was to an extent fuelled by the appearance on the market of proprietary codecs working on the public telephone network at lower data bitrates than those targeted by H.261. The next chapter detailed the improvements made to block based coding particularly within the ITU-T H.263 [15] and H.263+ [24] activities. The core H.263 algorithm made several key optimisations to H.261 including sub-pixel accuracy motion vectors and a new transform coefficient coding scheme with events more suitable to the more highly quantised events of low bit rate coders. The more significant enhancements came from other more complex, and hence optional, techniques including new PB picture types, relaxation of motion vector constraints, overlapped block motion compensation and arithmetic entropy coding. The author's work in this area involved development and incorporation of the syntax based arithmetic coding proposal into the common ITU-T simulation software, further research into adaptive versions of SAC, and investigation of a new proposal from MPEG concerning intra block prediction. The contribution to higher coding efficiency of each of these techniques by themselves was small. In combination, however, the full H.263 algorithm achieved over twice the compression performance of H.261 and for a modest 50% estimated extra complexity.

8.1.2 Content Coding Developments

Improved compression performance over H.263 was never demonstrated by SIMOC. Fundamental problems in this algorithm, particularly within the image analysis stages, limited further development here. The content coding ideas within SIMOC, however, and the more abstract model of video as projections of real world regions was seen to offer potential both to compression and more radically the idea of content based video manipulation. The work of Haavisto and Niewglowski [21,22] employing shape coding and affine regional motion models did, in fact, show efficiency gains over H.263 at a later stage. The main driver, though, for the emerging ISO/IEC MPEG-4 [37] multimedia coding standard, described in the next chapter, was content based functionality and this became the focus for the rest of the thesis research.

MPEG-4 was an ambitious project with a somewhat ill-defined focus for much of its early phase of development. Higher compression was the single initial target but H.263 was also seen to be addressing this very effectively. The content based functionality focus was not initially seen as a real industry need but did offer new content production and consumption paradigms and was in the end well supported by over 200 companies drawn from the telecommunications, broadcasting and multimedia industries. The standard made considerable progress in this newly adopted content based video coding field defining standardised techniques for arbitrary shape and texture coding in addition to extending the definition of content to include synthetic video, animated face and body models, synthetic audio and text and graphics. This was combined with the best block-based video compression tools of H.263 to define a very flexible and extensible content-based video standard.

Although MPEG-4 fully supported techniques for coding shape and alpha transparency planes, the means to generate this information for the content to be coded was intentionally not specified. The generation of this so-called *segmentation* information was considered a producer or encoder issue. MPEG philosophy had always been that only decoder issues should be specified to guarantee inter-working; enabling competition between companies as to which can provide the best encoder engine, and also, in this instance, provide the most useful segmentation data. A large amount of producer material can be created now to take advantage of content based MPEG-4 coding. Blue-screen, or chroma keying, techniques are commonly used in everyday weather forecasts and news bulletins and more often now in television special effects departments and simple binary shape data is readily available from this source. Computer games or virtual reality applications should also allow easy object shape extraction from the predefined graphics models. It was, however, foreseen that the benefits of content based coding and manipulation would not be readily realised for generic video. Content definition and extraction from generic video is an easy process for most of the animal kingdom. It is a very difficult task, though, for current generation computers, relying on complex image analysis and artificial intelligence techniques to perform even the most basic content analysis.

8.1.3 Content Analysis – Techniques and Frameworks

Chapter 6 detailed the author's research into one analysis technique known as feature clustering which aimed to build pixel feature vectors made up of multiple image cues such as colour, texture and motion and combine them in an appropriate feature space clustering algorithm. This chapter also examined the use of this approach in tracking of content through whole sequences. The aim of this research work was to outline a simple but flexible clustering strategy that could be applied to image and video features for content definition and tracking purpose. Success was achieved with simpler image feature vectors than originally thought necessary i.e. 3 component colour and 2 component position information only.

Chapter 7 detailed the current state of the COST 211^{quat} *Analysis Model* (AM)[60], a common software simulation model, which represented a concerted European effort to combine techniques of content definition and tracking specifically aimed at taking advantage of developments in the standards arena. Also detailed was a comparison of the image partitioning algorithms that could be used within the AM and a review of the MPEG-4 VOP generation frameworks to automatically extract content of interest from generic video. Complete content analysis frameworks capable of inferring content-of-interest from generic video whilst completely free of human intervention are the holy grail of this problem area and have proved as difficult to find. The difficulties here are imposed by the nature of the original content and the fact that all analysis is restricted to work only with the information available. We must not forget that the system we are trying to emulate here, the human visual system, has stereo vision, less spatial and temporal sampling problems and supporting acoustic information to work with. In contrast our analysis frameworks are very limited. The problem statement is also ill defined. The content of interest is a user dependant quality and will always depend on some degree of user interaction. Frameworks that provide the most effective user assistance are the more realistic focus now and this is the area the Analysis Model and the various MPEG-4 VOP generation algorithms try to address. Several alternative frameworks are still being examined by the COST 211 group, however, the group is also collaborating on the development of one modular framework with parallel low level pixel analysis processors computing region segmentations before a final rule driven composite segmentation. The pixel processing modules themselves vary in complexity and definition performance but essentially perform the same function, that of simplifying the input data to a set of useful regions ready for later rule based combination.

8.2 Recommendations for Future Work

8.2.1 Compression

Considerable progress has been made in video compression to the point now where acceptable video can be transmitted over even the most restricted public switched and mobile channels. Efforts of a considerable number of researchers have effectively optimised a range of generic, standardised, and low-complexity techniques. Further optimisations, while possible, will likely incur significant complexity increase for minimal return on coding efficiency. Having said that, efforts are continuing within the H.263 long term group with the H.263L project investigating some new algorithms and approaches.

The object based coding approach still shows promise to inch compression upward but work must be concentrated on the analysis phase of this approach – the weakest link in the SIMOC coder examined here.

Further more dramatic progress in this area may also be possible within the MPEG-4 Synthetic/Natural Hybrid Coding domain and the more specific applications of human head and body modelling and processing of other synthetically generated video.

8.2.2 Content Analysis

8.2.2.1 Feature Clustering

Extension of the feature clustering procedure to higher dimensional feature vectors would not be a problem. Texture [80], optic flow motion [77], and depth information from stereoscopic imagery or structure-from-motion work [53] are also readily available candidates for future development of the initial segmentation stages. For cluster tracking in subsequent frames, closed region size and principal components of the shape radii [56, 57] could also prove useful.

The clustering process itself also relies on a certain amount of supervision, at least on initial sequence frames, to select the best initial conditions, distance metric, feature component scaling and cluster merge threshold. Pauwels et al. [58] proposes a fully unsupervised algorithm making no assumption about the underlying data distribution. Their work uses a particular convolution kernel, a difference of Gaussian kernel, allowing for better discrimination between clusters. The width of this kernel is estimated from the data using partial sample cross validation. Interestingly this kernel has a positive centre and a negative surround, a similar characteristic structure of the centre-surround receptive fields found in retinal ganglion cells. This would also be a useful area for future development.

The clustering segmentation proposal also remains a bottom up segmentation process relying on a human observer to classify resulting regions. For semantic extraction and automatic characterisation of video sequence content it is necessary to combine this approach with a model or expectation based process. One route forward here is to supply the critical user interaction component with the provision of a user interface allowing labelling of content of interest regions. Such a scheme is employed in the work of O'Connor et al. [76]. This scheme also augments the mouse- driven frame marking of single region pixels with fine watershed data.

8.2.2.2 Analysis Model

The AM rule based approach has proved effective but the definition of the rule system here is critical. We risk forcing a pre-defined notion of content-of-interest if these rules are wrong. For example, the change detection module in the AM has a specific notion of one static background object and mostly singleton moving foreground objects. This was also the model for the SIMOC object based compression encoder and can work well for specific scenes but should not be imposed on other scenes for which it makes less sense. Even motion generally, whilst important, can not be the sole differentiator of content of interest. An accurate segmentation may still be required even when an object stops moving. Virtual avatar applications, as an example, representing real but segmented people and re-compositing in a shared environment, would not ideally want chunks of real texture suddenly disappear as soon as the real people sat down. This is addressed to some extent within the Change Detection AM module with the memory concept but it is more of a change in the rule system that is required emphasising, for the virtual avatar example, more texture and shape coherency than common motion.

Another argument that can be made, however, is that we can not clearly specify these rules as a predefined list or even apply them to a predefined hierarchy of partial segmentation results. The qualities that make something the content-of-interest at a certain moment may not be totally obvious even to the user. Appropriate interfaces, perhaps with smart user agents able to dynamically modify rules and weightings of attributes, will be a key component of any final analysis model framework.

Several criticisms are levelled at the AM and other MPEG-4 VOP generation frameworks in Chapter 7. However, they all represent a significant advance in our ability to derive benefit from content-based video coding from existing and generic video sources.

8.3 General Conclusions

The compression and content analysis work reported on in this thesis has been successful in evaluating and extending the limits of current techniques in computer vision and video coding for both high video compression and content based video functionality. Contributions have been made to the key developments in both these areas. If these lead to a better understanding and perhaps further breakthroughs in the brave new world of smart multimedia content then the pursuit of this research will have been very worthwhile.

References

1 ITU-R Recommendation BT.601, "Encoding parameters of digital television for studios"

2 Ziv J. and Lempel A., "A universal algorithm for sequential data compression", IEEE Transactions on Information Theory, 23, pp 337-343 (May 1977).

3 Harashima H., Aizawa K. and Saito T., "Model Based Analysis Synthesis Coding of Videotelephone Images - Conception and Basic Study of Intelligent Image Coding", IEICLE Transactions, 72, 5, pp 452-459 (May 1989).

4 Welsh W.J., Ph.D. Thesis, "Model-Based Coding of Images", University of Essex (1991)

5 Rydfalk, M., "Candide : a parameterized face", Research report of Dept. of Elec. Eng., Linkoping University, Sweden

6 Forchheimer, R. et al., "A semantic approach to the transmission of face images" Picture Coding Symposium 1984, 10.5, 1984

7 Wang, J.Y.A., Adelson, E.H., Desai, U., "Applying Mid-level Vision Techniques for Video Data Compression and Manipulation", Proceedings of the SPIE, Digital Video Compression on Personal Computers: Algorithms and Technologies, vol. 2187, San Jose, February 1994.

8 Musmann, H.G., Hotter M., Ostermann, J., "Object-oriented analysis-synthesis coding of moving images", Image Communication, 1, No. 2, pp 117-138, (Oct 1989)

9 ISO/IEC JTC1/SC29/WG11/N2564, "MPEG-4 Overview", Rome, December 1998.

10 ISO/IEC JTC1/SC29/WG11/N2571, "MPEG-7 XM Development", Rome, December 1998.

11 Beaumont J.M., Ph.D. Thesis, "A neural-based approach to Image Segmentation", Imperial College, (1996)

12 Huffman D., "A method for the construction of minimum redundancy codes", Proc. IRE, pp 1098-1101 (1962)

13 ITU-T Recommendation H.261, "Video codec for audio-visual services at p x 64 kbit/s"

14 International Standard IS 11172-2, "Coding of moving video and associated audio at rates up to about 1.5 Mbit/s, Part 2: Video"

15 ITU-T Recommendation H.263, "Video coding for narrow telecommunications channels"

16 Nagao, M., "Picture recognition and data structure." In Nake, Rosenfeld (Eds.) Graphic Languages., North Holland, 1972.

17 Ekman P., Friesen, W.V., "Facial Action Coding System", Consulting Psychologists Press, Palo Alto CA., 1977.

18 Bierling M., "Displacement Estimation by hierarchical block matching", SPIE Visual Communications and Signal Processing, 1001, (1988)

19 COST211^{ter} Simulation Subgroup, "Simulation model for Object Based Coding: SIMOC-1", SIM(94)31, (1994)

20 European COST211 Group, "Redundancy Reduction Techniques and Content Analysis for Multimedia Services", http://www.teltec.dcu.ie/cost211

21 Contribution LBC-96-091, "Proposal for Advanced Video Coding", ITU-T SG15 Low Bitrate Coding Experts Group, April 22-26, 1996.

22 Haavisto P. et al., "Proposal for Efficient Coding", ISO/IEC JTC1/SC29/WG11 MPEG96/0904, Tampere, July 1996.

23 Stiller C., "Motion-Estimation for Coding of Moving Video at 8kbit/s with Gibbs Modeled Vectorfield Smoothing", SPIE Vol. 1360 Visual Communications and Image Processing 1990, pp 468-476.

24 ITU-T H.263+ Latest Draft, ftp://standard.pictel.com/video-site/h263plus/draft21.zip

25 Mitchell J.L., Pennebaker W.B., "Software implementation of the Q-coder", IBM Research Report, RC 12660, (April 1987)

26 International Standard IS 10918-1 and ITU-T Recommendation T.81,"Information technology - digital compression and coding of continuous tone still images, Part 1: Requirements and guidelines".

27 Penbaker W.B. and Mitchell J. L., "JPEG still image data compression standard", Van Nostrand Reinhold (1992)

28 ITU-T Recommendation H.245, "Control protocol for multimedia communication"

29 Azadegan F., "Use of Inter-Block Compression to Improve Coding Efficiency of Intra-Coded Frames", ISO/IEC JTC1/SC29/WG11 MPEG96/0623, Munich, January 1996.

30 Bjontegaard G., "Results from intra core experiment T9/T10 and comparison with DC-prediction", ISO/IEC JTC1/SC29/WG11 MPEG96/0986, Tampere, July 1996.

31 Mulroy P., "Results of core experiments on improved intra picture coding T9/T10", ISO/IEC JTC1/SC29/WG11 MPEG96/0918, Tampere, July 1996.

32 Tan T.K., Shen S.M., "Intra Prediction (T9/T10) and DC/AC Prediction Results", ISO/IEC JTC1/SC29/WG11 MPEG96/0939, Tampere, July 1996.

33 Tan T.K., Shen S.M., "Results for Core Experiment T9 - DC/AC Prediction", ISO/IEC JTC1/SC29/WG11 MPEG96/1178, Chicago, October 1996.

34 ISO/IEC JTC1/SC29/WG11/N2203, "Information Technology - Coding of Audio-Visual Objects: Audio, ISO/IEC 14496-3, Final Committee Draft", Dublin, July 1998.

35 ISO/IEC JTC1/SC29/WG11/N2202, "Information Technology - Coding of Audio-Visual Objects: Visual, ISO/IEC 14496-2, Final Committee Draft", Dublin, July 1998.

36 ISO/IEC JTC1/SC29/WG11/N2201, "Information Technology - Coding of Audio-Visual Objects: Systems, ISO/IEC 14496-1, Final Committee Draft", Dublin, July 1998.

37 ISO/IEC JTC1/SC29WG11/N2323 "MPEG-4 Overview (Dublin Version)", Dublin, July 1998.

38 Virtual Reality Modelling Language (VRML) Consortium, http://www.vrml.org

39 Mortlock A., Machin D., McConnell S., Sheppard P., "Virtual conferencing" BT Technol J, 15, No. 4, pp 120-129 (October 1997)

40 Welsh W.J., Searby S and Waite J B : "Model-Based image coding", BT Technol J, 8, No. 3, pp 94-106 (July 1990)

41 ITU-T Recommendation T.82 / JBIG, "Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-Level Image Compression"

42 Bossen F., Ebrahimi T., "A simple and efficient binary shape coding technique based on bitmap representation", ISO/IEC JTC1/SC29/WG11 MPEG96/0964, Tampere, July 1996.

43 Brady N., "Adaptive Arithmetic Encoding for Shape Coding", ISO/IEC JTC1/SC29/WG11 MPEG96/0975, Tampere, July 1996.

44 Sikora, T., "The MPEG-4 video standard verification model", IEEE Transactions on Circuits and Systems for Video Technology, 7 (1), 19-31, 1997.

45 Sikora T., Makai B.,"Shape-Adaptive DCT for Generic Coding of Video", IEEE Trans. Circuits and Systems for Video Technology, Vol. 5, No. 1, Feb. 1995.

46 ISO/IEC JTC1/SC29WG11/N2552 "MPEG-4 Visual VM v.12", Rome, December 1998.

47 Rose S., "The Conscious Brain", Weidenfeld & Nicolson, 1973, pp 47.

48 Gregory R., "Even Odder Perception", (Routledge), 1994, Chapter 8, pp 53.

49 Jain, A K., Dubes, R.C., "Algorithms for Clustering Data", (1988)

50 Tou J.T., Gonzalez R.C., "Pattern Recognition Principles", Addison-Wesley, 2nd edition (1977)

51 Serra J., Vincent L., "An overview of morphological filtering", Circuits, Systems and Signal Processing, vol. 11, no. 1, pp 47-108, January 1992

52 Weldon T.P., Higgins W.E., Dunn D.F., Gabor filters, Pattern Recognition, 29:12, 2005-2015 (1996)

53 Azarbayejani A., Horowitz B., Pentland A., "Recursive estimation of structure and motion using relative orientation constraints", IEEE CVPR Conference (1993)

54 Mulroy P.J., "Spatial and Temporal Image Segmentation by Feature Clustering", Intl. Workshop on Coding Techniques for Very Low Bitrate Video, VLBV '95, Tokyo, Nov. 1995.

55 Mulroy P.J., "Clustering as a Video Object Plane formation tool", COST 211ter Simulation Subgroup, SIM(96)04, Barcelona, February 1996.

56 Gupta L., Srinath M.D., Pattern Recognition, 20:3, 267-272 (1987)

57 Kashyap R.L., Chellapp R., IEEE Trans. on Information Theory, 27:5, 627-637

58 Pauwels E., Fiddelaers P., Van Gool L., "Unsupervised Clustering Using Centre-Surround Receptive Fields with Applications to Colour-Segmentation", Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'97, pp.143-148, Louvain-la-Neuve, Belgium, June 1997.

59 Marichal X., De Vleeschouwer C., Macq B., "Toward Visual Search Engine Based on Fuzzy Logic", Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'97, pp.135-141, Louvain-la-Neuve, Belgium, June 1997

60 COST 211ter Simulation Subgroup, "Description of the COST211 AM (Version 4.0)", SIM(98)14, Dublin, July 1998.

61 Alatan A.A., Tuncel E., Onural L., " A Rule-based Method for Object Segmentation in Video Sequences", Proceedings of IEEE ICIP 97, Santa Barbara, CA, vol. II, pp. 522-525, October 1997.

62 Morris O.J., Lee M.J, Constantinides A.G., "Graph Theory for Image Analysis: an Approach Based on the Shortest Spanning Tree", IEE Proceedings, vol. 133, pp 146-152, April 1986.

63 Mulroy P.J., "Video Content Extraction : Review of Current Automatic Segmentation Algorithms", Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'97, pp.45-50, Louvain-la-Neuve, Belgium, June 1997

64 Aach T., Kaup A., Mester R., "Statistical Model-based Change Detection in Moving Video", Signal Processing, vol. 31, no. 2, pp. 165-180, March 1993.

65 Alatan A.A., "Object-based 3-D Motion and Structure Analysis for Video Coding Applications", Ph.D. Dissertation, Bilkent University, February 1997.

66 COST 211^{quat} Analysis Model, Web pages, http://www.teltec.dcu.ie/cost211/am/am.htm

67 Burt P.J., Hong T-H, Rosenfeld A., "Segmentation and Estimation of Image Region Properties through Co-operative Hierarchical Computation", IEEE Trans. Systems, Man, and Cybernetics. Vol 11, No. 12, pp 802-209, December 1981.

68 Beaumont J.M., "Image Segmentation based on a Neural Model", Ph.D. Thesis, Imperial College of Science, Technology and Medicine, June 1996.

69 Czerepiński P., Bull D.R., "Morphological Methods for Image and Video Coding: An Overview", Chapter 22: Insights into Mobile Multimedia Communication, Academic Press (1999).

70 Vincent L., Soille P., "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol 13, No. 6, June 1991.

71 Correia P., Pereira F., "Segmentation of Video Sequences in a Video Analysis Framework", Proceedings of Workshop on Image Analysis for Multimedia Interactive Services WIAMIS '97, pp. 155-260, Louvain-la-Neuve, Belgium, June 1997.

72 ISO/IEC JTC1/SC29WG11 MPEG97/2702, "Description of automatic segmentation techniques developed and tested for MPEG-4 Version 1", Fribourg, October 1997.

73 Colonnese S., Neri A., Russo G., Tabacco C., "Adaptive Segmentation of Moving Object versus Background for Video Coding", Proc. of SPIE Annual Symposium, vol. 3164, San Diego, August 1997.

74 Mech R., Wollborn M., "A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera", Proceedings of Workshop on Image Analysis for Multimedia Interactive Services WIAMIS '97, pp. 57-62, Louvain-la-Neuve, Belgium, June 1997.

75 Choi J.G., Kim M., Lee M.H., Ahn C., "Automatic segmentation based on spatio-temporal information", ISO/IEC JTC1/SC29WG11 MPEG97/2091, Bristol, April 1997.

76 O'Connor N.E., Brady N., Marlow S., "Supervised Image Segmentation using EM-Based Estimation of Mixture Density Parameters", Proceedings of Workshop on Image Analysis for Multimedia Interactive Services WIAMIS '97, pp. 27-32, Louvain-la-Neuve, Belgium, June 1997.

77 Horn B.K.P., Schunk B.G., "Determining optical flow", Artificial Intelligence, vol. 17, pp 185-203 (1981)

78 Wang J.Y.A., Adelson E.A., "Spatio-temporal segmentation of video data", Technical Report No. 262, MIT Media Laboratory Vision and Modelling Group, February 1994

79 Diehl N., "Object-oriented motion estimation and segmentation in image sequences", Signal Processing: Image Communication, vol. 3, no. 1 pp 23-56 (1991)

80 Tuceryan M. "Moment Based Texture Segmentation", Procs 11th IAPR Intl. Conf. on Pattern Recognition, Vol. III, Conf. C : Image, Speech and Signal Analysis, The Hague, NL, IEEE, Aug. 1994

81 ISO/IEC JTC1/SC29WG11 M0974, "Adaptive Arithmetic Coding (JBIG Coder) of DCT Coefficients", Tampere, July 1996

82 ISO/IEC JTC1/SC29/WG1 N205, "Lossless and Lossy Compression of Text Images by Soft Pattern Matching"

83 Nowlan, S.J., "Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures", CMU-CS-91-126, School of Computer Science, Carnegie Mellon University, Pitsburgh, PA. (1991)

84 Bottou L., Bengio Y., "Convergence properties of the K-Means Algorithms", Proc. of NIPS94 - Neural Information Processing Systems: Natural and Synthetic, 28 Nov. - 3 Dec. 1994, MIT Press pp 585-92.

85 Dempster A.P., Laird N.M. and Rubin D.B., "Maximum-likelihood from incomplete data via the EM Algorithm", Journal of Royal Statistical Society B, 39:1-38. (1977)

86 Xu L., Jordan M., "Theoretical and experimental studies of convergence properties of the em algorithm for unsupervised learning based on finite mixtures", Presented at the Neural Networks for Computing Conference, 1994.

87 Chen C.W., Luo J., Parker K.J., "Image Segmentation via Adaptive K-Mean Clustering and Knowledge-Based Morphological Operations with Biomedical Applications", IEEE Trans. on Image Processing, Vol. 7, No. 12, Dec. 1998.

88 Konrad J., Dubois E., “Bayesian Estimation of Motion Vector Fields.”, IEEE PAMI 14(9), September 1992.

89 Kearney J., Thompson W.B, Boley D.L., “Optical flow estimation: An error analysis of gradient based methods with local optimisation.”, IEEE PAMI, pages 229-243, March 1987.

90 Nagel H., Enkelmann W., “An investigation of smoothness constraints for the estimation of displacement vector field from image sequences” IEEE PAMI, pages 565-592, September 1986.

Appendix A : List of Publications with Appended Papers

IMPROVED MOTION VECTOR FIELDS FOR OBJECT-ORIENTED CODING USING LOCAL CONFIDENCE MEASURES
Mulroy P, Whybray M, Very Low Bitrate Video workshop, **VLBV '94**, Colchester

USE OF SMOOTHER MOTION VECTORS WITHIN SIMOC-1
Mulroy P, **COST 211ter Simulation Subgroup**, SIM(94)56, Berlin

RESULTS OF CORE EXPERIMENTS ON IMPROVED INTRA PICTURE CODING T9/T10
Mulroy P, ISO/IEC JTC1/SC29/WG11 MPEG96/0918, Motion Picture Experts Group**, MPEG**, July 1996, Tampere

SPATIAL AND TEMPORAL IMAGE SEGMENTATION BY FEATURE CLUSTERING
Mulroy P, Very Low Bitrate Video workshop, **VLBV '95**, Tokyo

CLUSTERING AS A VIDEO OBJECT PLANE FORMATION TOOL
Mulroy P, **COST 211ter Simulation Subgroup**, SIM(96)04, Feb 1996, Barcelona

VIDEO CONTENT EXTRACTION : REVIEW OF CURRENT AUTOMATIC SEGMENTATION ALGORITHMS
Mulroy P, Workshop on Image Analysis for Multimedia Interactive Services, **WIAMIS '97**, Louvain-la-Neuve

VRML GETS REAL THE MPEG-4 WAY
Mulroy P, **IEE Colloquium**, Teleconferencing Futures, June 1997, London

VIDEO CODING - TECHNIQUES, STANDARDS AND APPLICATIONS
Whybray M, Morrison G, Mulroy P, **BT Technology Journal**, Oct 1997

Improved motion vector fields for object oriented coding using local confidence measures

P J Mulroy, M W Whybray
BT Laboratories
Visual Applications Division
Martlesham Heath, Ipswich, UK
email : mulroy_p_j@bt-web.bt.co.uk

INTRODUCTION

Motion estimation within object oriented coding typically uses a 3-level hierarchical block matching algorithm based on Bierling's work [1]. This algorithm is an advance on previous ones which used fixed measurement window sizes however it does not exploit any local contextual information or inter-block motion correlation which could lead to higher accuracy in estimated vectors and reduce overall vector field entropy. This paper proposes the use of such information to associate a degree of confidence with each motion estimate thereby allowing vectors of dubious accuracy to be influenced by neighbouring ones judged to be good predictors, with the aim of producing a more stable and accurate vector field. This area has been explored to a degree by Stiller [2] and his suggested smoothing criterion has been implemented within the context of an object based coding scheme. The coding scheme used is based on the COST211 SIMOC1 model, which in turn is derived from work at the University of Hannover [3].

CURRENT PROCEDURE

Bierling's algorithm involves a 3 stage hierarchical process of block matching and when applied to object oriented coding a degree of global contextual information is also typically used in so far as the algorithm is only applied to changed regions as defined by the pre-computed change detection mask. The stage parameters used are set out below in the following table :

Stage	Window size	Step size
1	32x32 window on mean value filtered data	±2 pels first then ±1 pel
2	16x16 window on original unfiltered version	±1 pel
3	16x16 window on bilinearly interpolated versions of original (8x8 window in original sampling grid)	±1 pel (± 0.5 pel step in original)

Within our coding scheme the search is performed at every 16th pel in every 16th line resulting in individual motion vectors of ± 4.5 pel resolution. These vectors are then bilinearly interpolated to provide a value for the vector field at every pel in the object. As it stands this approach alone has a number of problems. The vector range chosen is quite restrictive when it is considered that our scheme also codes at a reduced frame rate and performs temporal sub-sampling. It is equivalent to a maximum of ±1.5 pel per frame of a 25Hz sequence and while this is mostly adequate for the 'Miss America' test sequence a more challenging and active sequence such as 'Susie' poses a major problem. As with most hierarchical schemes false lock conditions can arise depending on search positions chosen, and as each vector is used in interpolating adjacent vectors an incorrect estimate can adversely affect a large area of the synthesised image. This in turn will increase the number of regions which fail to fit the flexible 2D object model used, and increase the required bit-rate.

The criteria for selecting the best block match at each stage of Bierling's scheme is the lowest mean absolute difference (MAD) value. If these MAD values are considered to form an energy surface for each search position then the aim is to demonstrate a correlation between the characteristics of these energy surfaces and the associated confidence of the chosen 'best' vector in corresponding image regions. In areas of low image detail, such energy surfaces can be shallow and this would indicate the likelihood of false matches using the conventional approach and resulting therefore in an unstable motion vector field. On the other hand regions with a high degree of textural information will result in energy surfaces with better defined minima, and the standard minimum search should lead to a vector which would have a high degree of confidence associated with it. This research tests these assumptions and reports on results from the Miss America test sequence. One smoothing approach we propose is to impose a flexible grid on node points of the estimated motion vector field and allocate stiffness constraints to branches of this grid. The particular stiffness constraint for any particular branch would be determined by the confidence measure associated with that node point which in turn would be based on evaluation of local contextual and textural information and adjacent motion vectors. A number of next closest minima positions could be extracted from the shallow surface regions and these positions tested for associated vector coherence using perhaps a vector field model similar to that proposed in [2]. Results of this approach are not conclusive however as our confidence measures are not reliable. Latest results will be shown at the workshop.

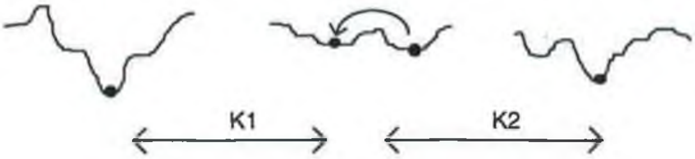


Figure 3 - *1D representations of mean absolute difference energy surfaces*

Figure 3 above illustrates the principle applied on 1 dimensional data. The first and third samples show deep minima suggesting confident vector matches. The second sample exhibits local minima however and an inappropriate minimum could be chosen depending on the search pattern. Use of the proposed method could result, depending on context, in relatively high stiffness constraints K1 and K2 forcing the vector to a more coherent and accurate position. Due to the expected reduction in vector field entropy a denser field could also be generated leading hopefully to a closer approximation to the true movement in the sequence.

RESULTS

Stiller [2] proposed a smoothing operation based on minimising a cost function of two models, namely a displacement model which also addressed camera noise and a vector model which favoured a vector field of quasi-stationary segments. Figures 1 to 5 show results of our implementation of this approach. The initial estimates to the smoothing algorithm are based on minimum mean absolute difference as calculated for a full ±8pel search on 16x16 pel blocks, rather than using a hierarchical search. The overall vector field entropy is reduced after smoothing and spurious vectors such as one within the right shoulder region are seen to be corrected.

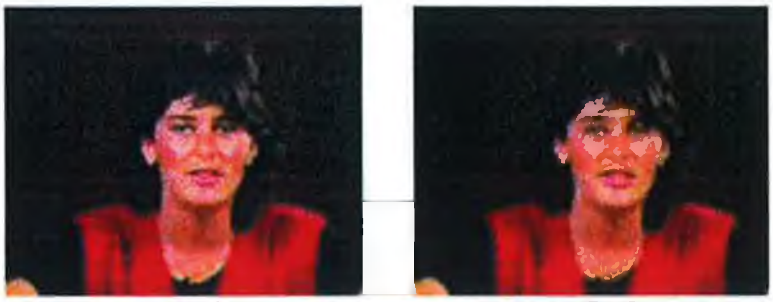


fig. 1 *fig. 2*

Miss America : Frames 78 & 81 of original test sequence.

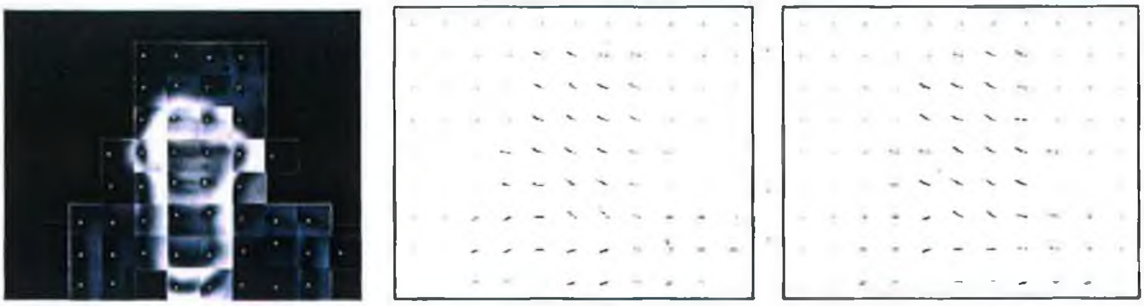


fig. 3 *fig. 4* *fig. 5*

Associated mean absolute difference surfaces, motion vector field and field after smoothing

The actual motion in this scene pair is down and to the right. Vectors are calculated with respect to the previous frame and are reversed. Figure 3 shows the associated MAD surfaces and the minimum points of these surfaces are denoted by a single white pel. It is the nature of these surfaces that was examined as a contender for a reliable confidence estimate which could be used as an extension to Stiller's model.

The hessian operator was used in examining the correlation between high detail regions and their MAD surfaces. This operator is defined as

$$H(f) = \frac{d^2f}{dx^2}\ \frac{d^2f}{dy^2} - (\frac{d^2f}{dxdy})^2$$

This tends towards a maximum in areas of high luminance change in both horizontal and vertical directions. It is for this reason that it is used extensively in feature location algorithms. Tables 1 & 2 below show results from a ± 8 pel full search on 16 x 16 blocks of raw image data. The image data in this case is from frames 90 and 93 of Miss America.

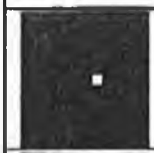
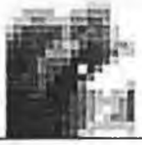
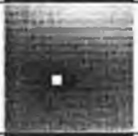
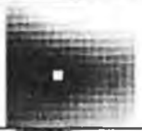
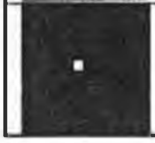
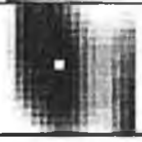
MAD Surfaces	Histogram Equalized	Average Image Hessian	Maximum Image Hessian
		0.3	4
		0.2	5.4
		0.2	1.9

***Table 1** : Mean absolute difference surfaces associated with low image hessian blocks*

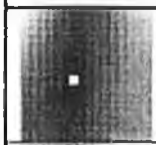
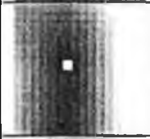
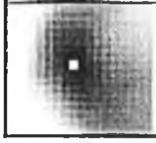
MAD Surfaces	Average Image Hessian	Maximum Image Hessian
	9.9	107.7
	18.3	143.4
	11.1	212.7

***Table 2** : Mean absolute difference surfaces associated with high image hessian blocks*

The examples do broadly comply with original assumptions as to the nature of surfaces, with regions of low average hessian resulting in shallow energy surfaces and high hessian regions giving better defined minima.

However, the surfaces were found to be far more dependent on the nature of the detail in the block rather than the amount of that detail. The second example in table 2 has the highest average hessian in the frame shown and corresponds to the boundary of hair with the left side of Miss America's face. This has a sharp boundary and hence high hessian but only in one direction and results in a valley shaped minimum region on the MAD surface. This means that the horizontal component of the motion vector is well defined, but the vertical component is not and a false value may easily result.

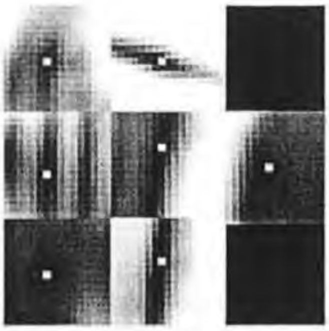


fig. 6

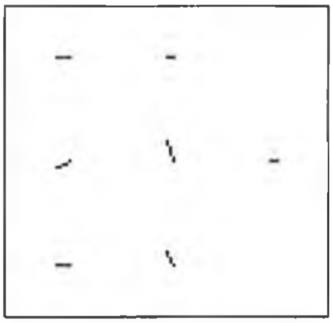


fig. 7

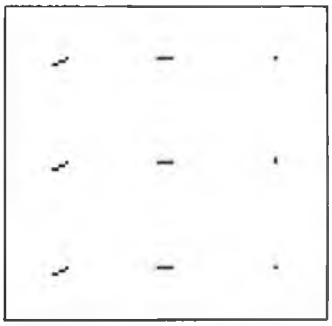


fig. 8

Set of MAD surfaces of shoulder region from later frame pair and vector fields before and after smoothing

Figure 6 above shows a set of low hessian value surfaces which have been histogram equalised to better reveal the shallow structure. Figure 7 shows the initial estimates passed to the smoothing algorithm and figure 8 the result of smoothing with a high weighting given to the vectorfield model. The central vector is seen clearly to move along the surface minimum valley to a more coherent vector position and illustrates the benefit of this approach.

Figure 6 also illustrates the existence of local minima which can lead a hierarchical block matching method in the wrong direction at an early stage, and prevent it finding the true lowest minimum. Although it has been shown above that this lowest minimum is not always the best one, it is likely to be a better starting point for an iterative smoothing operation than a secondary minimum.

CONCLUSIONS

Two characteristics of block matching MAD error surfaces which can lead to poor vector fields have been identified. One is the existence of local minima which can lead hierarchical search type motion estimators in the wrong direction at an early stage, and prevent location of the true minimum. The second is the common occurrence of valley shaped minimum regions, which leave the motion vector poorly defined along the direction of the valley. The hessian operator was found to identify deep minima, but further work is needed to recognise and deal correctly with the valley shaped features. Stiller's method has been shown to be an effective means of correcting erroneous vectors in general, although it may be improved by selecting secondary minima as candidate alternative vector positions, rather than just some function of the neighbouring vectors as he proposed [2], and by modifying the weighting between the displacement and vector field cost functions dependent upon measures of the nature of the MAD error surfaces.

REFERENCES

[1] M. Bierling, "Displacement Estimation by hierarchical block matching", SPIE Visual Communications and Signal Processing, Vol. 1001, (1988).
[2] C. Stiller, "Motion-Estimation for Coding of Moving Video at 8 kbit/s with Gibbs Modeled Vectorfield Smoothing", SPIE Visual Communications and Image Processing, Vol. 1360, (1990).
[3] H.G. Musmann, M. Hotter and J. Ostermann. "Object-oriented analysis-synthesis coding of moving images", Signal Processing: Image Communication, Vol. 1, No. 2, (October 1989), pp117-138.

COST 211 ter simulation subgroup **SIM(94)56**
Berlin, 20-21 October 1994

SOURCE: **BT Laboratories**
TITLE: **Use of smoother motion vectors within SIMOC-1**
PURPOSE: **Information**

1. INTRODUCTION

This report details the results obtained using smoother motion vector fields within the object based analysis synthesis coding scheme proposed in SIMOC-1 [1]. A smoothing algorithm first proposed by Stiller[2] has been implemented and tested on a range of motion vector grid sizes. Comparison is made with the existing hierarchical block matching scheme and some conclusions drawn. The relative merits of denser vector grids over coarser ones, as looked at by SIM(94)39 [3], are also discussed.

2. SMOOTHING ALGORITHM

Motion vector fields for image coding are typically generated using block matching between successive frames of an image sequence. The criteria used in such schemes is the minimum mean absolute difference between displaced blocks. Stiller has proposed an extension of this using both a displacement model of minimum mean absolute difference and a vectorfield model which favours segmentwise smooth vectors. The cost function which he uses is derived in [2] and set down here.

$$\mathrm{C} = N \ln(\max(mad, N.P_c)) + \sum_{n=1}^{8} \frac{c}{2l}.||v - v_n||$$

N number of pels in block

mad mean absolute difference of changed pels within displaced block

P_c Power of camera noise estimated from image

c Vectorfield model weighting

l distance between considered pels (l=1 for four nearest, √2 for diagonal neighbours)

v test vector

v_n neighbouring vectors

This is applied iteratively on the initial motion vector field, three times in total and the vectors giving the minimum cost at each iteration kept. The test vector candidates are listed below :

- vector calculated by the previous estimation step
- eight neighbours calculated by previous estimation step
- four vectors differing from the first candidate by 1/2 pel
- weighted average of the eight neighbour vectors applying weights 1/l
- predicted vector of the predictive vector coding

Some modifications were made to this in our implementation. In the object based scheme the change detection mask is available and can be used in the vectorfield model of the cost function. We need only consider the neighboring vectors if they are within this mask. This would allow discontinuities in the vector field at the mask boundaries where they should be expected to be.

A second modification made was in the choice of test vectors. As the arithmetic coding used in SIMOC-1 does not use the same prediction mechanism as Stiller's coder the prediction test vector was omitted.

The modifed cost function used therefore is given below :

$$C = N \ln(\max(mad, N.P_c)) + \sum_{n=1}^{8} M.\frac{c}{2l}.\|v - v_n\|$$

M Change detection mask (M=1 if v_n footpoint within change detection mask, else M=0)

and the test vectors chosen are :

- vector calculated by the previous estimation step
- eight neighbours calculated by previous estimation step
- four vectors differing from the first candidate by 1/2 pel
- weighted average of the eight neighbour vectors applying weights 1/l

The effect of smoothing in itself can be seen in figs 2.1 and 2.2 where the generated field is smoother with less spurious vectors and the field entropy can be seen to be reduced. The contention is that this more closely matches the real motion in the scene while also reducing the bit rate for motion data and this is what was tested here.

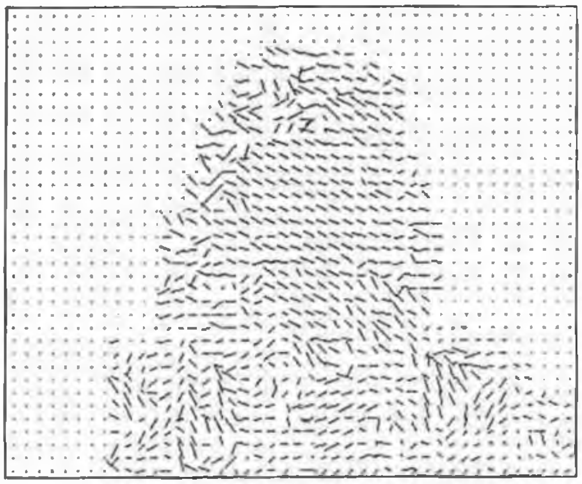


Fig 2.1 - Vectorfield generated between frames 75 & 78 of original Miss America Sequence

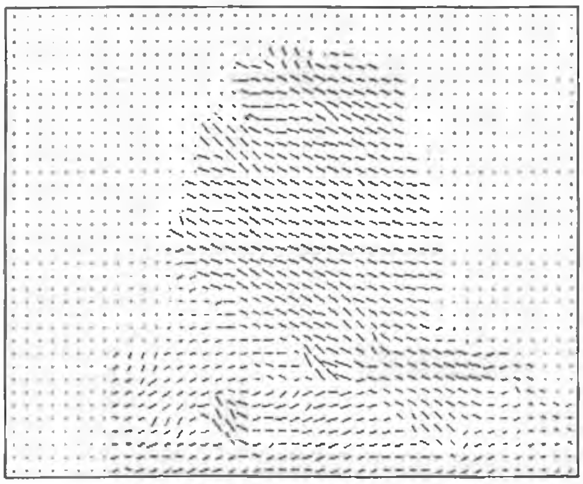


Fig 2.2 - Smoothed vectorfield of identical image pair.

3. RESULTS

The first set of results show bit counts for the motion and colour parameters for the test sequence *Miss America* coded using SIMOC-1 at three different vector grid densities and three different extremes of smoothing applied to the vectorfield. 50 frames of *Miss America* at a framerate of 8.3Hz were used.

3.1 Varied grid sizes

3.1.1 Results for 16 x 16 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits
No smoothing	361	12666	13461
Smoothing, c=20	346	13074	13863
Smoothing, c=80	271	13804	14489

Table 3.1 - Average bit counts per frame

3.1.2 Results for 8 x 8 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits
No smoothing	1538	6398	8487
Smoothing, c=20	1223	8085	9783
Smoothing, c=80	737	13055	14203

Table 3.2 - Average bit-counts per frame

3.1.3 Results for 4 x 4 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits
No smoothing	6612	1599	8797
Smoothing, c=20	3551	3155	7281
Smoothing, c=80	1586	15868	17808

Table 3.3 - Average bit-counts per frame

In the implementation of SIMOC under test the motion parameters are coded as in the specification document [1] with a simple previous block prediction followed by adaptive arithmetic coding. The vectors themselves are calculated using the hierarchical block matching algorithm with ± 4½ pel range, again specified in [1], however the block sizes used are proportional to the grid density chosen. The colour parameters are coded using DPCM rather than vector quantisation as specified but results should hold for VQ coding too.

Some early conclusions can be made on the basis of the initial set of tests :

- For all grid densities the vectorfield's entropy is reduced when smoothing, as expected, and a lower bit rate for the motion parameters is achieved. This is more evident for the denser field tests as can be seen in figs 3.3 to 3.6. Table 3.3 shows a reduction from over 6000 bits with no smoothing applied through to 1500 bits with the most extreme smoothing factor used.

- Any modification of the motion vectors requires that we move away from the minimum error condition in a frame difference sense. It follows that the energy of the resulting prediction error image can only increase by using the smoothed vectors. This in turn increases the amount of model failure area and a higher bitrate for the colour coding and a higher overall datarate for the same quality is the end result. This also holds for the more dense motion vector fields although as mentioned above the reduction in bits due to motion is more marked in these cases.

- Also, although giving the best reduction in bits for motion, the smoothing case with c=80 is found to be far too extreme in general. Fig 3.6 shows a fivefold increase in the colour parameter bitcount for the same overall quality. Disregarding this most extreme smoothing case then the reduction in motion parameter bit-rate due to smoothing ranges from ~50% to ~5% depending on vector density.

Subjectively the sequences coded with or without smoothing show little difference for this set of results and the luminance SNR measures are also very similar. This is due mainly to the model failure detection method, defined in [1], which 'targets' a certain overall quality, in this case an SNR of ~36dBs, and which is similar for all tests in this section.

3.2 Varied grid sizes with constrained model failure area

The second set of results relate to a modified SIMOC encoder where a model failure area constraint mechanism has been used to limit the final datarate. The mechanism used restricts the number of MF pels to a finite level - 1000 pels in this case which equates to 4% of the image. This was found to work well for the test sequence used but is not generally applicable. Its purpose here was twofold, firstly to reduce the bitrate to below 32Kbit/s and secondly to counteract the increase in prediction error signal due to smoothing and allow the motion information to more directly influence the quality of the output image.

3.2.1 Results for 16 x 16 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits	Avg. Y-SNR
No smoothing	356	1399	2317	34.91
Smoothing, c=20	326	1406	2303	34.90
Smoothing, c=80	255	1387	2155	34.72

Table 3.4 - Average bit-counts per frame and luminance SNR in dBs

3.2.2 Results for 8 x 8 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits	Avg. Y-SNR
No smoothing	1562	1414	3538	36.12
Smoothing, c=20	1169	1324	3043	35.82
Smoothing, c=80	724	1406	2633	34.63

Table 3.5 - Average bit-counts per frame and luminance SNR in dBs

3.2.3 Results for 4 x 4 grid

	Avg. motion bits	Avg. colour bits	Avg. total bits	Avg. Y-SNR
No smoothing	6634	1571	8806	37.47
Smoothing, c=20	3555	1450	5549	36.69
Smoothing, c=80	1524	2027	3968	30.14

Table 3.6 - Average bit-counts per frame and luminance SNR in dBs

The second set of tests give a clearer view of the effect of motion information within SIMOC when model failure detection cannot be relied upon to cater for significant distortion errors. The effects can be readily seen in the comparitive SNR measures detailed in Sect 3.3.

The different MF detection mechanisms used between the two sets of tests also throws up an apparent anomaly in the colour parameter bitcount measures which should be noted. In the original MF detection process, with a defined model compliant variance threshold, less colour parameter bits are required for the initial coded frames than in the constrained MF case. That is because with the MF pel limit constraint the assumption was made that more pels than this limit would be classified as model failure to maintain overall quality. This is valid in all but the first few frames of the sequence where although 1000 MF pels are allowed and used, less colour bits are required. Some tests were carried out using both the variance threshold and the MF-pel limit to cater for this case but the effect to average bit-counts and SNRs was found to be minimal.

3.3 Comparative SNR Measures

Fig. 3.13 illustrates the direct influence of grid density on picture quality. These results are for constrained MF area with no smoothing applied and the increase in bit-rate is due solely to coding the extra motion vectors. It can be seen however that on doubling the vector grid density an average increase of 1-2 dB gain in SNR is achievable.

If we attempt to use bit-rate as the constraint, and allow coarser grids more model failure area, we get the results shown in fig. 3.14. This still shows an improvement using denser grids but it is now marginal (½ - 1dB). Here the efficiency of coding of the motion parameters is the issue.

Focussing on the use of smoothing again and the most likely implementation of this within SIMOC, fig. 3.15 shows what is achievable at the same bitrate using a smooth 8x8 motion vector grid rather than the specified 16x16 grid. A small improvement in SNR (½-1dB) is evident.

If with improved arithmetic coding we can code the denser but smoother grid of motion vectors with the same bitrate as an unmodified 16x16 grid, as is suggested in [2], we could see better improvement. Fig. 3.16 illustrates this case.

4. CONCLUSIONS

The source model within SIMOC-1 has proved to be extremely reliant on good motion estimation for successful operation. A sparse vectorfield generated using hierarchical block matching does provide a generally reliable estimate but this can be dramatically improved upon by using fields of higher density. As has been shown in this report denser fields with better overall motion estimates can be realised without necessarily incurring large bit-rate increases. Marginal improvement in decoded picture quality can be shown now at the same bitrate but better predictive arithmetic coding suggests more significant improvements of 1-2dB gain in signal to noise ratio could be made at the same overall bitrate.

5. FURTHER WORK

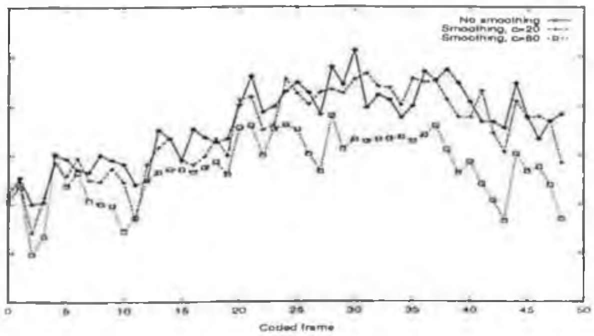
One major candidate for further development in this area is the search for a more efficient motion parameter coding algorithm. The current method detailed in SIMOC-1 uses a simple previous block prediction mechanism which could be improved upon. The author of [2] details a predictor based on weighted neighbours and this is hailed by him as the best choice. This should be investigated. Also, different methods of arithmetic coding could be employed.

REFERENCES

[1] DCU (editors), "SIMOC1", COST 211ter Simulation Subgroup, SIM(94)31, BOSCH, UNI-HAN, BT Labs, RWTH Aachen, CNET, NTR, KUL, PTT, LEP Philips, EPFL, Siemens, CSELT, DBP-Telekom, Bilkent University, UPC, HHI, IST.

[2] C. Stiller, "Motion-Estimation for Coding of Moving Video at 8kbit/s with Gibbs Modeled Vectorfield Smoothing", SPIE Vol. 1360 Visual Communications and Image Processing '90, pp 468-476.

[3] Daimler-Benz Research Centre Ulm, "On the Drawbacks of a Coarse Grid for Motion Estimation", COST 211ter Simulation Subgroup, SIM(94)39.



3.1 Motion parameter bitcounts : 16x16 grid

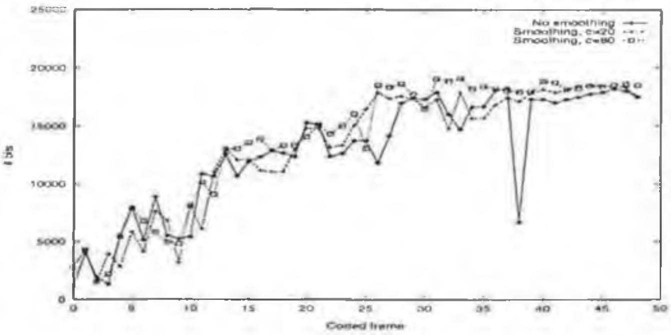
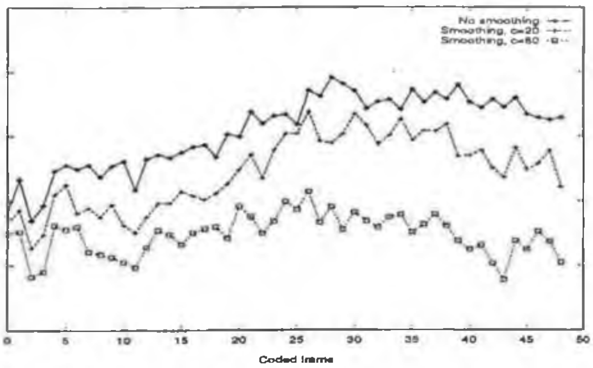


fig 3.2 Colour parameter bitcounts : 16x16 grid



g 3.3 Motion parameter bitcounts : 8x8 grid

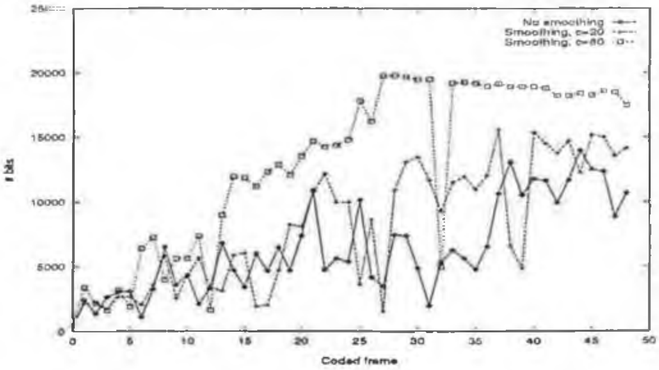
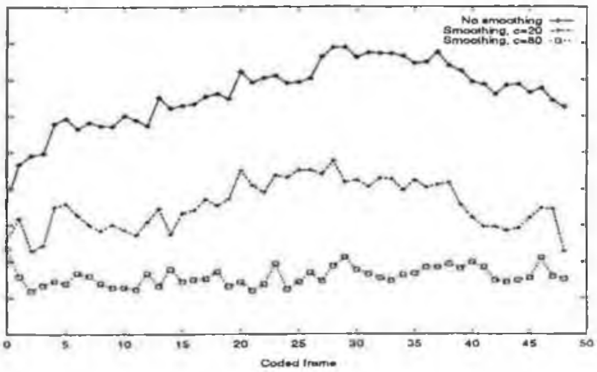


fig 3.4 Colour parameter bitcounts : 8x8 grid



g 3.5 Motion parameter bitcounts : 4x4 grid

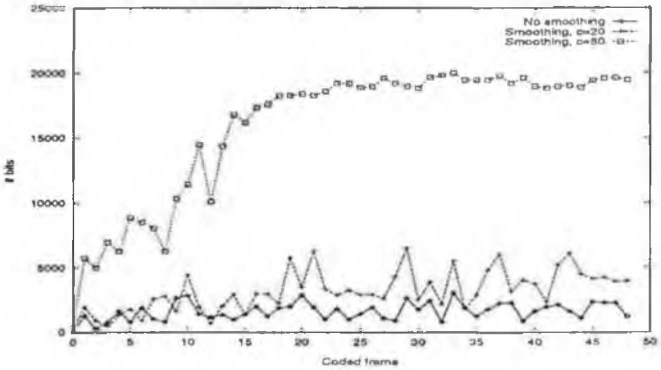
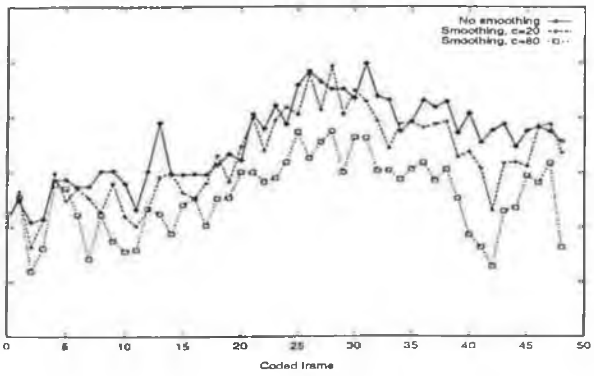


fig 3.6 Colour parameter bitcounts : 4x4 grid



3.7 Motion parameter bitcounts : 16x16 grid with constrained model failure area

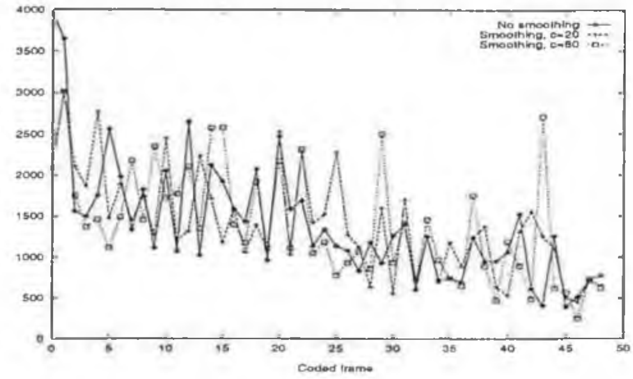
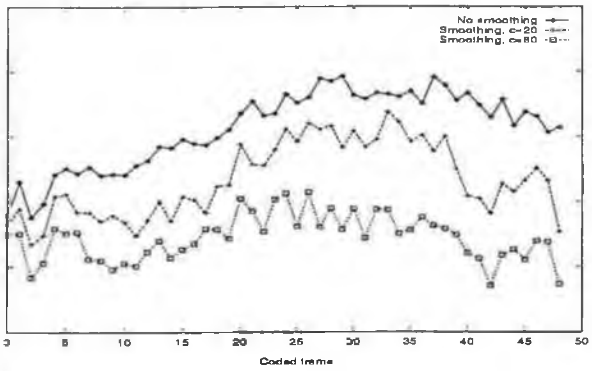


fig 3.8 Colour parameter bitcounts : 16x16 grid with constrained model failure area



3.9 Motion parameter bitcounts : 8x8 grid with constrained model failure area

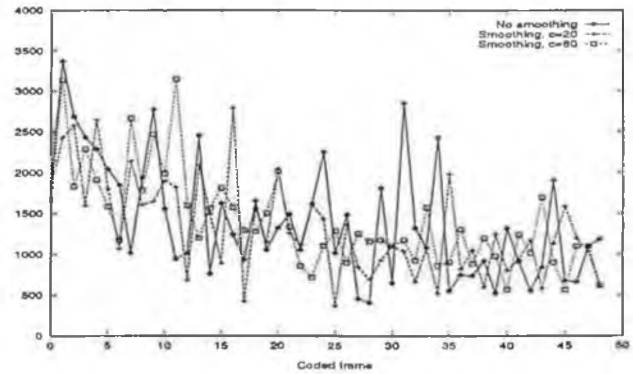
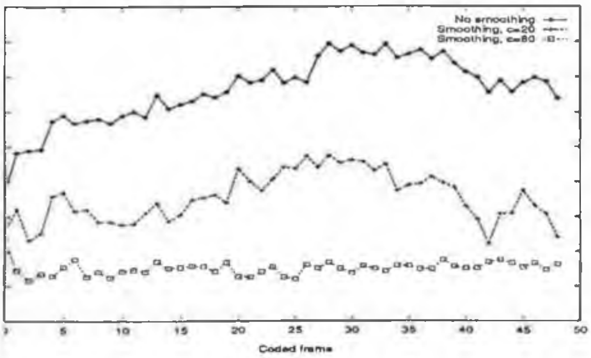


fig 3.10 Colour parameter bitcounts : 8x8 grid with constrained model failure area



3.11 Motion parameter bitcounts : 4x4 grid with constrained model failure area

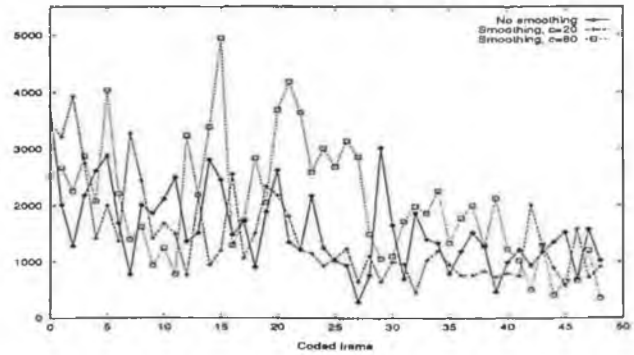
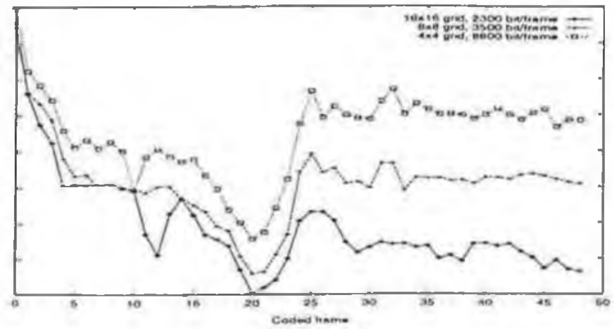
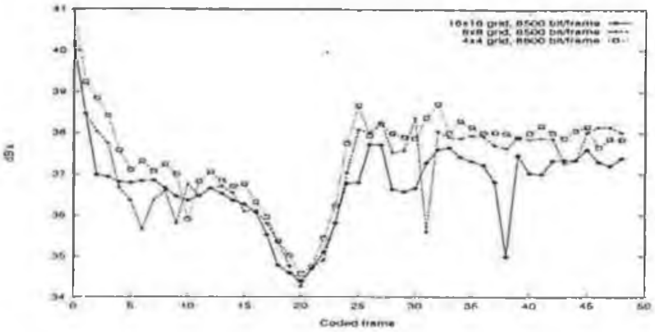


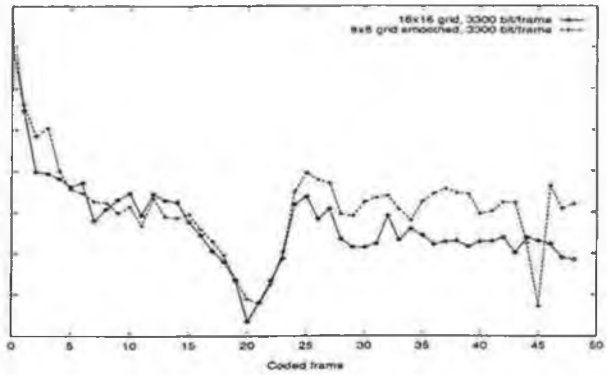
fig 3.12 Colour parameter bitcounts : 4x4 grid with constrained model failure area

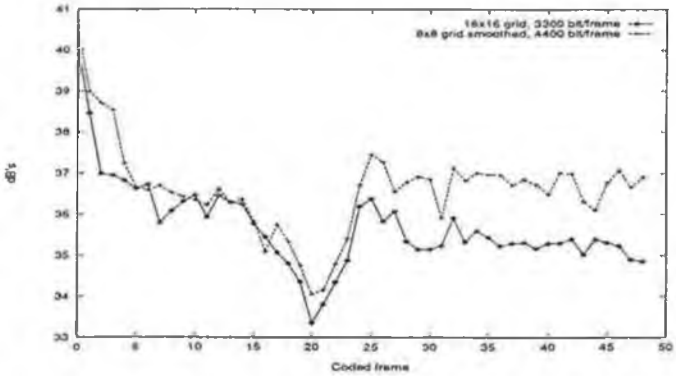




'.13 Luminance SNR comparison of vector grid densities with constrained MF area

fig 3.14 Luminance SNR comparison of vector grid densities with constrained bit-rate





.15 Luminance SNR comparison of 16x16 grid and smoothed 8x8 grid with same bitrate

fig 3.16 Luminance SNR comparison of 16x16 grid and smoothed 8x8 grid with same MF area

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG96/0918
July 1996

Source: **BT Laboratories, UK**
Status: **Report**
Title: **Results of core experiments on improved intra picture coding T9/T10**
Author: **Patrick Mulroy**

1. Introduction

Comparison is made to H.263 (the VM in single frame VOP mode) and also to MPEG-1 intra coding where DC component prediction is used. The decoded results show, for most test frames, a consistent SNR and subjective gain for the Telenor/GTE proposal over both the non-predictive H.263 intra coder and the DC predictive MPEG-1 intra coder. The gain is higher for larger quantiser values and the results are shown in rate distortion curves. A D1 tape comparison will also be shown at the meeting.

2. Test Conditions

The core experiment description is given in [1] and is not repeated here. The experiment was implemented within the latest version of the Telenor H.263 code (tmn v. 1.7) rather than the VM due to time restrictions. The VM in the mode under test defaults to H.263 so this was not considered a problem.

The new intra coding scheme can choose four different coding modes for each macroblock and two extra bits of syntax per macroblock are needed to signal which mode is chosen. These extra bits are taken into account in the results shown.

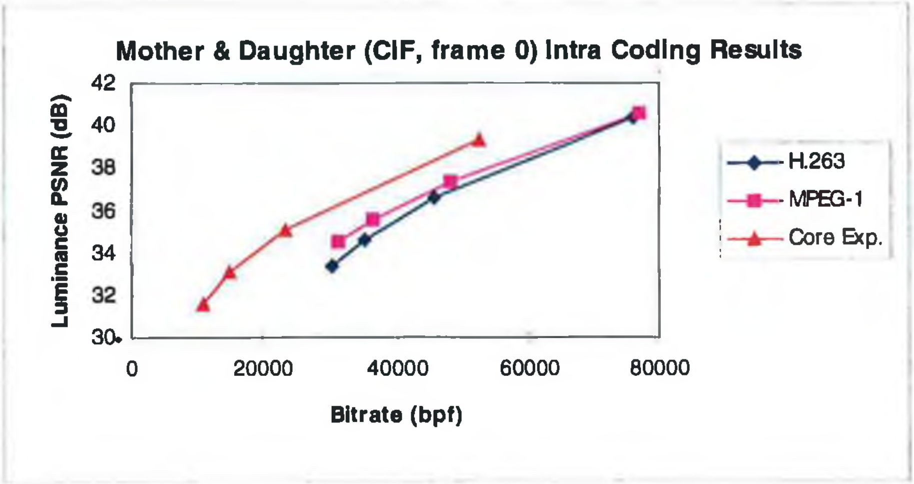
Results are obtained for the CIF first frames of the following test sequences - mother and daughter, hall monitor, coastguard, silent voice and news - and the QCIF first frames of - hall monitor, coastguard, silent voice and news.

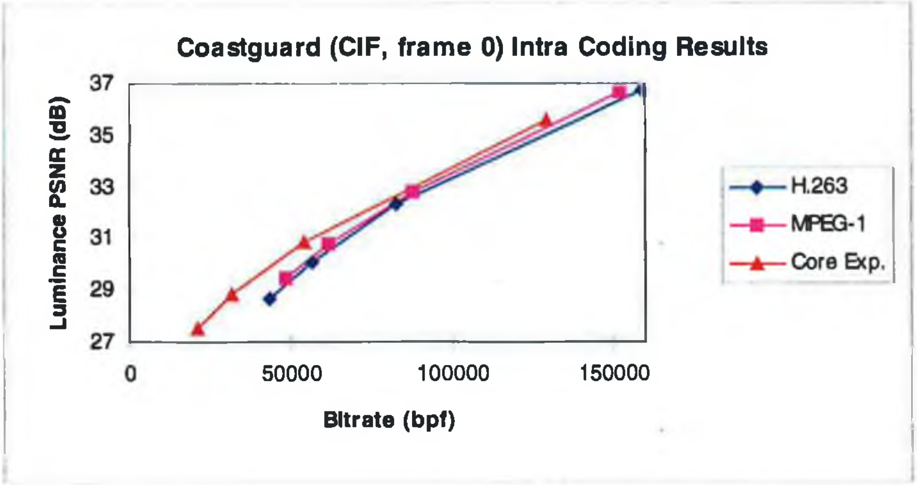
The D1 tape will show results of coding at the same bitrate between the the VM and the core experiment and also show bitrate savings for subjectively equal quality codings.

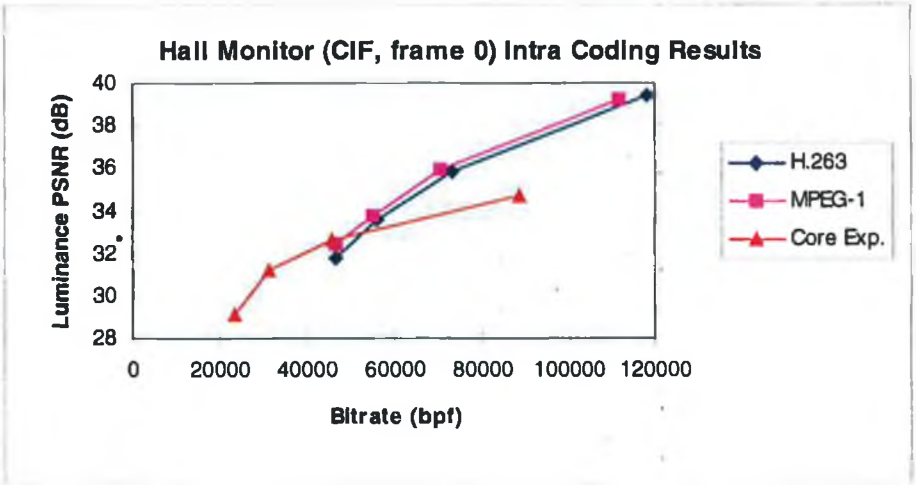
3. Test Results

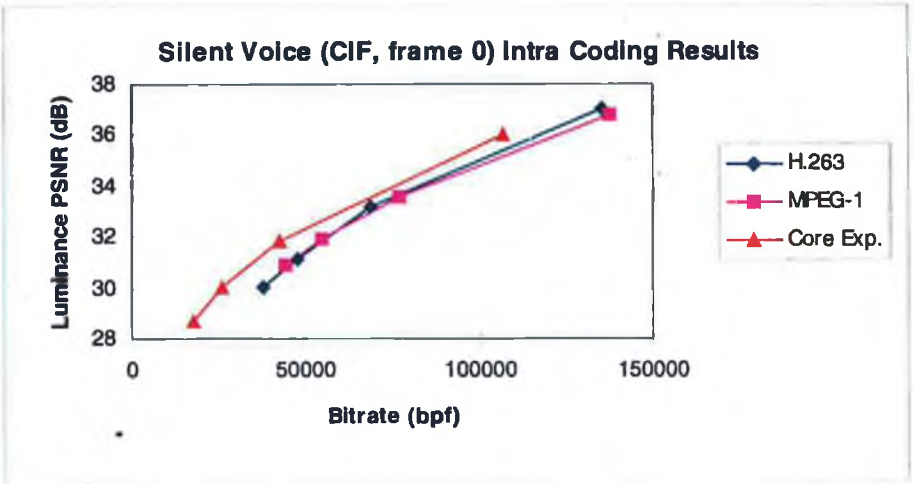
The results of coding with H.263, MPEG-1 and the core experiment are shown in the following rate distortion graphs. The luminance PSNR and bitcounts are plotted for quantisation parameter values of 5, 10, 15 and 20.

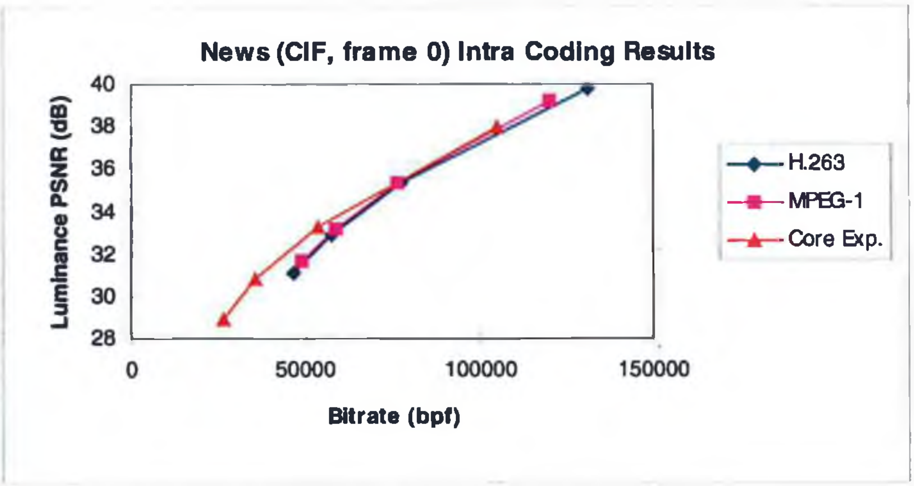
3.1 CIF Results



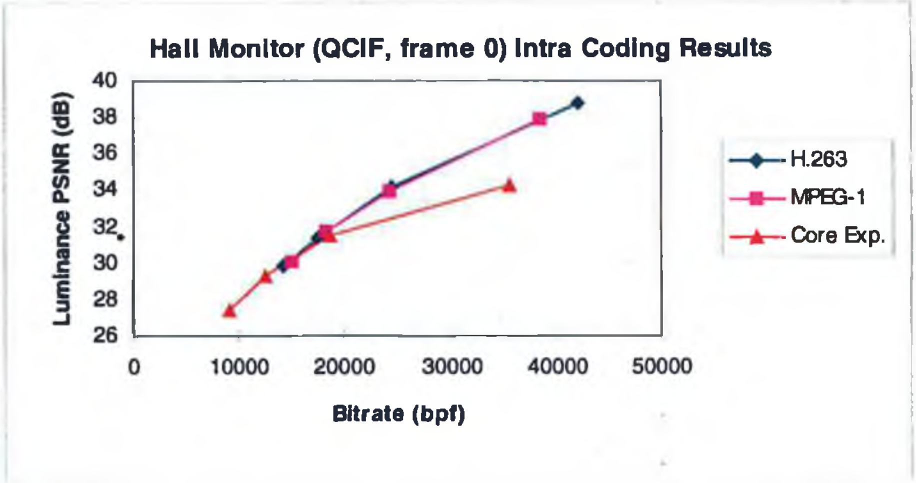


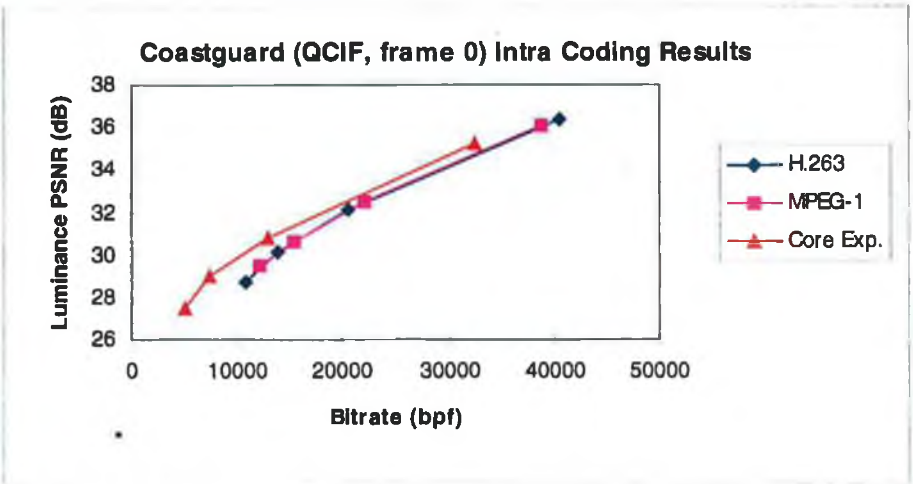


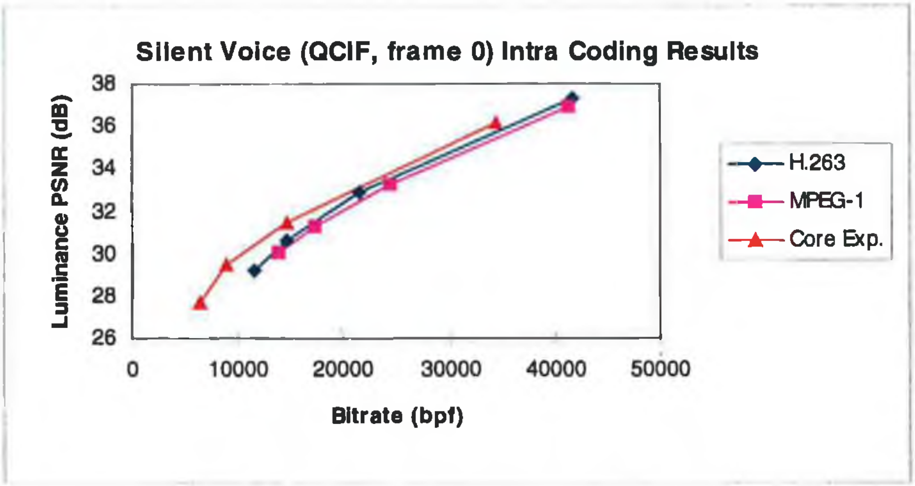


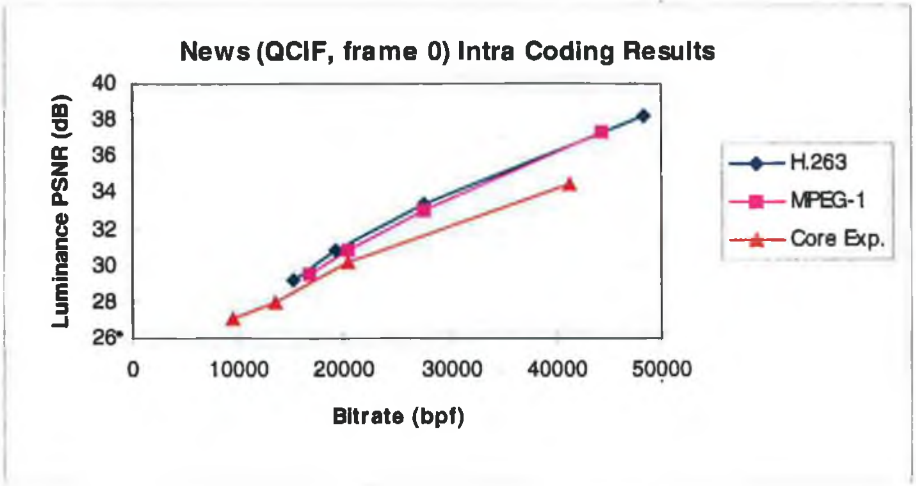


3.2 QCIF Results









4. Conclusions

The test results show a gain both objectively and subjectively for the majority of the intra frames tested. The exceptions are Hall Monitor (CIF and QCIF) and News (QCIF) where the core experiment performs worse particularly at the lower quantisation values. It is not immediately clear why this is the case but the use of the threshold suggested in [1] could at least improve these situations by falling back to the standard H.263 intra mode.

References

[1] ISO/IEC JTC1/SC29/WG11 N1250, Core Experiment Description for improved intra coding T9/T10.

Spatial and Temporal Image Segmentation by Feature Clustering

Patrick Mulroy
Centre for Human Communications,
BT Laboratories, Martlesham Heath, Ipswich, UK
email : pmulroy@visual.bt.co.uk

Introduction

Region based coding of moving video is currently under investigation as the principal contender for the next generation of coding techniques. It potentially offers support for very low bitrate video coding and content based functionality as envisaged by ISO/MPEG-4 [1]. Image segmentation is the first step in such an approach and is the subject of this contribution. Standard segmentation approaches range from histogram thresholding to 'watershed' techniques which typically operate on a single component or property of the scene (e.g. luminance level) and perform with variable success. Other approaches analyse multiple properties such as colour, texture and motion independently and then face the problem of combining multiple segmentations of the same scene. The approach presented in this paper is a two stage process whereby clustering is performed of features made up initially of only spatial luminance and chrominance information. These spatial clusters are then merged in the second stage by feature motion data derived from optic-flow analysis, the goal being to yield segmentations coherent in both motion and colour.

Clustering Procedure

Clustering of feature vectors based on an inter-feature distance metric is a widely used classification tool [2]. Feature vectors in the image feature context can be constructed from a set of individual pixel properties (e.g. luminance and colour difference), a set of neighbourhood properties (e.g. texture moments) or a combination of both. The use of 6-dimensional vectors derived from texture moments is detailed in [3].

In this work we have used relatively simple feature vectors of luminance and colour difference data {y u v}, in addition to pixel motion data derived from a coarse to fine optic flow algorithm. The clustering procedure used is based on the 'k-means' algorithm detailed in [2]. This algorithm is initialised with *k* arbitrary clusters and is designed to iteratively minimise the sum of the squared distances from all points in each cluster domain to the cluster centre. Each feature vector is allowed to change allegiance between clusters once during each iteration and the distance metric used is the *Mahalanobis distance* :

$$D = (\mathbf{x} - \mathbf{m})'C^{-1}(\mathbf{x} - \mathbf{m})$$

where *C* is the pooled sample covariance matrix, *m* is the mean vector and *x* is the vector under test. This measure exploits the statistics of the feature space and ensures scale-invariance of each vector component.

In our implementation the following steps were taken to perform the spatial clustering :

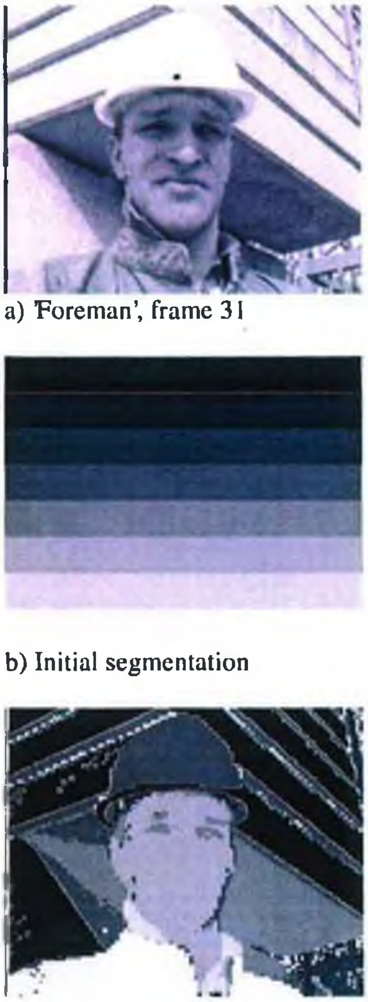
- Eight stripes were chosen as an arbitrary initial clustering of the feature vectors. A larger number of smaller initial clusters was tried but convergence was much slower.
- The covariance matrix, *C*, and its inverse were calculated for the entire feature space.
- For each vector its distances to all cluster centres were calculated and the vector reassigned to the closest one.
- The third step was repeated until the number of reassigned vectors dropped below a small threshold.

The resulting cluster regions could now be merged in their entirety in a further processing step making use of feature motion data and feature vectors made up of optic flow data. Use was made of an inter-cluster distance table and clusters of close proximity merged.

While the extension of these techniques to image sequences has not yet been fully tested the approach here is to repeat the above steps for subsequent frame pairs this time replacing the initial arbitrary segmentation with the previous frame pair segmentation.

Experimental Results

Several results are presented for the QCIF test sequence 'Foreman'. Figure 1 shows the raw clustering results for the spatial {y u v} data of frame 31 with an initial horizontal stripe segmentation which converges after 7 iterations to the spatial segmentation shown in Fig. 1c.



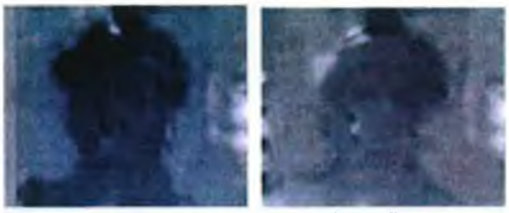
a) 'Foreman', frame 31

b) Initial segmentation

c) Final segmentation

Fig. 1 - {y, u, v} Feature Clustering

The above result does not in itself provide us with a means of grouping the segments into connected regions or objects. For this we need either to try to adapt the information to fit some model (e.g. a head and shoulders centrally located) or we need some extra information. Fig. 2 shows the x and y optic flow fields derived from a multi-scale coarse to fine optic flow algorithm described in [4] and [5] and applied to frame 31 and frame 33 of the test sequence.



a) x-flow b) y-flow

Fig. 2 - Optic Flow Fields

Two approaches using this optic flow information were assessed within our segmentation algorithm. The first and less successful approach involved incorporation of the optic-flow data directly into the feature space and clustering with 5 component feature vectors. The optic flow is not very accurate at edge boundaries and as it is given equal weight with the Mahalanobis clustering metric the resulting segmentation loses object boundary precision. The second more successful approach used the output of the iterative spatial clustering of three component {y u v} vectors as the initial segmentation input to a cluster merging process driven by the two component optic flow vectors alone. The resulting segmentation mask shown in fig. 3 is quite a good foreground and background segmentation while still retaining sharp object boundaries.

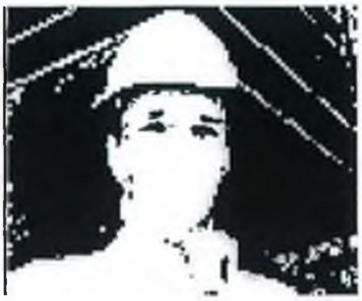
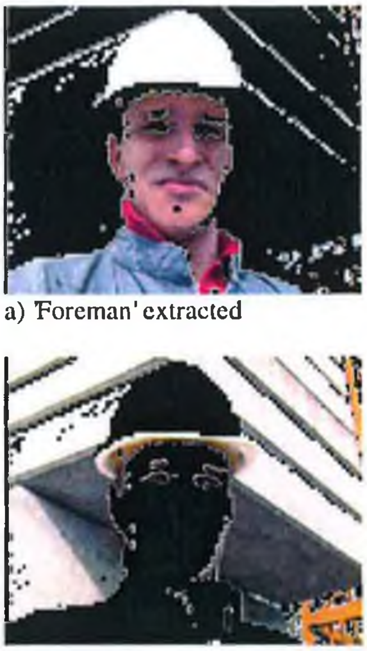


Fig. 3 - {y, u, v} clustering followed by {xflow, yflow} cluster merging

The result applied to the original image is shown in fig. 4. Although noisy, several methods could be employed to this mask to give a cleaner segmentation including small region reassignment and morphological processing. More in-depth post-processing techniques were not investigated but common approaches include lowpass filtering, relaxation or morphological operators. These techniques do tend to give more coherent segmentation results but region boundaries are usually less precise. One approach detailed in [6] tackles this problem by allowing an iterative feedback of smoothed regions back into the clustering process. Effectively a compromise can be reached between smooth homogeneous regions and precise boundary regions.



a) 'Foreman' extracted

b) Residue

Fig. 4 - Segmentation applied to input image with no morphological cleaning

Conclusions

Some early results have been presented which apply standard classification techniques to the problem of image segmentation. A useful segmentation for region and object based coders is considered to be one which is coherent in texture and motion and a procedure that could be used to achieve this is set out above. Several areas remain under investigation; type of feature vectors to use, use of localised texture moments as in [3], weightings of each feature vector component and type of final control to use for a particular type of segmentation. Finally the incorporation of this segmentation into an efficient coding scheme which can exploit this information is under study.

Acknowledgements

The author acknowledges the collaboration in this area with the video coding group at Dublin City University and the use of optic-flow software developed at the Media Lab, Massachusetts Institute of Technology.

References

[1] ISO/IEC JTC1/SC29/WG11 N0997, MPEG-4 Call for Proposals, July 1995, Tokyo.

[2] J. T. Tou and R. C. Gonzalez. Pattern Recognition Principles. Addison-Wesley Publishing Company, Inc., Massachusetts, 1974.

[3] M. Tuceryan. Moment Based Texture Segmentation, Procs. 11th IAPR Intl. Conf. on Pattern Recognition, Vol III, Conf. C : Image, Speech and Signal Analysis, Aug. 1994, The Hague, NL, IEEE.

[4] B. Lucas and T. Kanede. An iterative image registration technique with an application to stereo vision. Image Understanding Workshop, pp 121-130, 1981.

[5] J.Y.A Wang and E.H.Adelson. Layered representation for image sequence coding. IEEE ICASSP, volume 5, pp 221-224, Minneapolis, Minnesota, April 1993.

[6] O. Pichler et al. A Multichannel Algorithm for Image Segmentation with Iterative Feedback, Procs. 5th Intl. Conf. on Image Processing, July 1995, Edinburgh, UK, IEE

COST 211ter simulation subgroup SIM(96)04
Barcelona, 12-13 February 1996

SOURCE: P. Mulroy, BT Laboratories
TITLE: Clustering as a Video Object Plane formation tool
PURPOSE: Information

INTRODUCTION

The MPEG-4 Verification Model [1] supports representation of scene content by video object planes (VOPs) for subsequent coding and manipulation. For content based access applications VOPs with some semantic meaning are the most useful and this contribution investigates the use of image feature clustering to automatically extract the segments of interest from typical sequences. Results from this work as applied to still frames and frame pairs have already been presented [2] at VLBV '95. This work will show some preliminary results applied to sequences with positional and motion information used in the final clustering process. Similar work on *Automatic detection of interest areas* [3] has also been presented to COST211ter at the last meeting.

CLUSTERING PROCEDURE

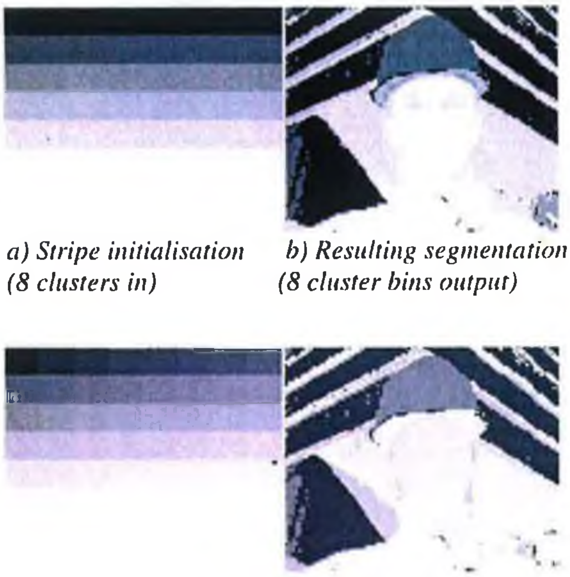
The clustering procedure for the image feature vectors is described in detail in [2] and reviewed briefly below. Relatively simple feature vectors of luminance and colour difference data {y u v} are used to perform the initial spatial image segmentations grouping regions of common colour. Motion and position data are then used to group these regions into semantic objects. The motion data is derived from a coarse to fine optic flow algorithm whereas the position criteria is a simple one grouping regions with a high percentage of their area within a particular region of the frame.

An implementation of the 'k-means' algorithm is used. This algorithm is initialised with *k* arbitrary clusters of feature vectors from the first frame and is designed to iteratively distribute samples among the cluster domains using a minimum distance classifier with new cluster centres calculated after each iteration. For the segmentation work the initial cluster centres were chosen to be feature vector means of stripes or blocks of the image frame. The distance metric used was the *Mahalanobis* distance which exploits the statistics of the feature space and ensures scale-invariance of each vector component. In a variation from the general k-means algorithm distances between cluster means are also measured at each iteration and very close clusters combined. Algorithm convergence for each frame was signalled by the number of feature reassignments dropping below a certain threshold (less than 10 pixels reassigned in this case). Fig. 1 shows some clustering results for both stripe and block initialisations. Subsequent frames were initialised with the cluster assignments from the previous frame and the algorithm run to convergence for all remaining frames.

EXPERIMENTAL RESULTS

Motion and position information was used separately after the initial spatial segmentation stage and figures 3 and 4 show the effect of the different criteria. For motion information a coarse to fine optic flow algorithm was used and figure 2a shows the x and y flow images between a frame pair within the 'foreman' sequence. This flow information was initially used directly in the feature clustering stage using 5 dimensional vectors but it was found to be more useful as a post clustering stage combining clusters of common or close average motion. Figure 3 shows the result of this process with 4 different frames of 'foreman' giving an instantaneous indication of regions with common motion.

Figure 2b shows the position mask used in a 'head and shoulders' extraction example. If 60% of the regions area was within the black portion of the mask the entire cluster was considered part of the area of interest. Figure 4 shows the application of this criteria alone, after an initial spatial clustering stage, to the 'mother and daughter' and 'foreman' test sequences. Sequences will also be shown at the meeting.



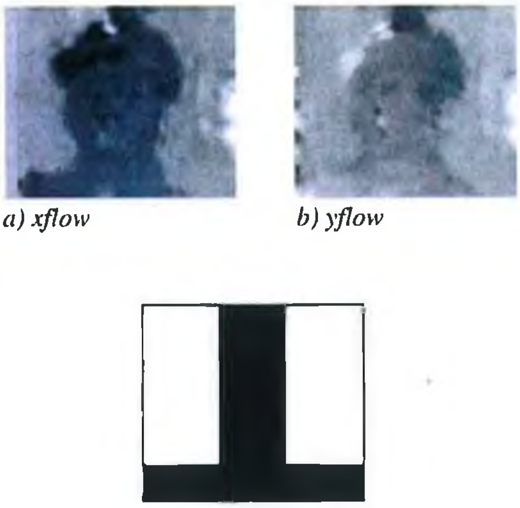
a) Stripe initialisation (8 clusters in)

b) Resulting segmentation (8 cluster bins output)

a) Block initialisation (64 clusters in)

b) Resulting segmentation (12 cluster bins output)

Figure 1 : Initialisations and resulting segmentations



a) xflow

b) yflow

c) position mask for 'head and shoulders' extraction

Figure 2 : Optic flow and position mask information

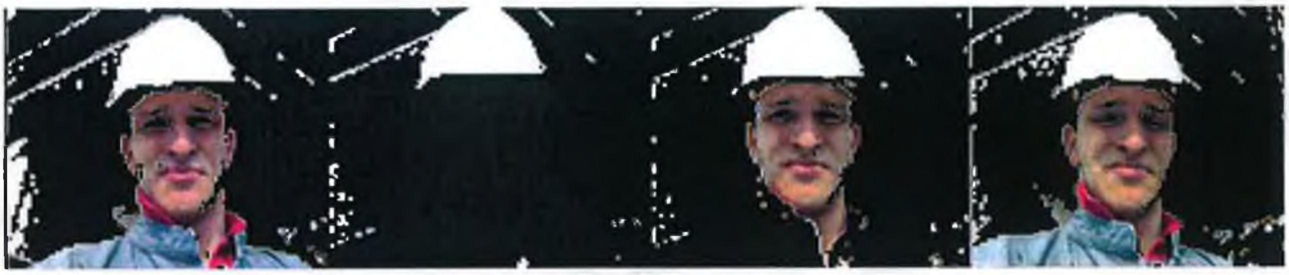
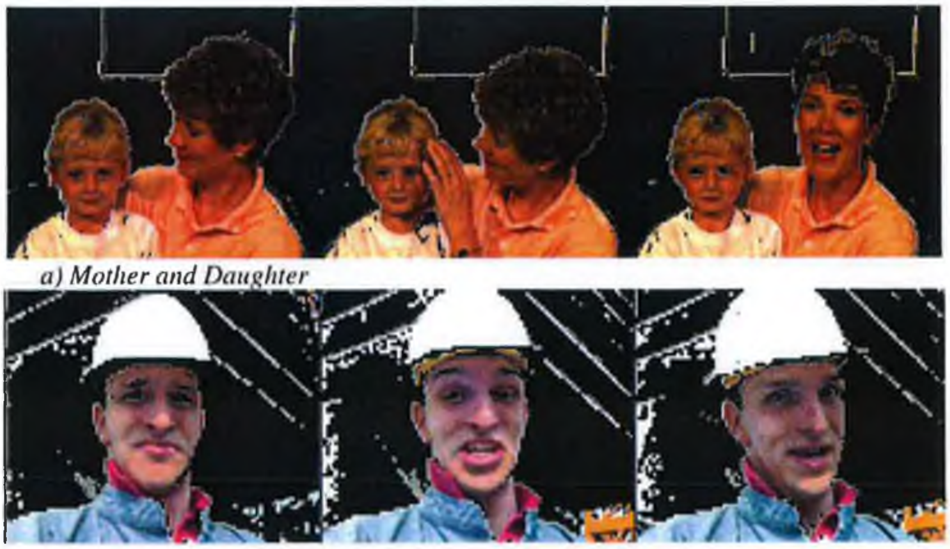


Figure 3 : Motion criteria extraction set



a) Mother and Daughter

b) Foreman

Figure 4 : Position criteria extraction set

CONCLUSIONS AND NEXT STEPS

Some early results have been presented which apply standard classification techniques to the problem of image segmentation and semantic object extraction with some initial promising results. The results are still quite noisy at this stage both spatially and temporally but post processing steps to avoid this have not been investigated as yet. The clustering process itself is also found to be quite dependent on the characteristics of the feature space and methods of improving this are also under investigation. Several other areas are also being examined; type of feature vectors to use, use of localised texture moments, weightings of each feature vector component and type of final control to use for a particular type of segmentation.

ACKNOWLEDGEMENTS

The author acknowledges the collaboration in this area with the video coding group at Dublin City University and the use of optic-flow software developed at the Media Lab, Massachusetts Institute of Technology.

REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 N1172, MPEG-4 Video Verification Model, Version 1.0, January 1995, Munich.

[2] P. Mulroy. Spatial and Temporal Image Segmentation by Feature Clustering, Very Low Bitrate Video Workshop, Paper I-5, November 1995, Tokyo.

[3] UCL. Automatic detection of interest areas, SIM(95)43, COST 211ter simulation subgroup, September 1995, Paris.

VIDEO CONTENT EXTRACTION : REVIEW OF CURRENT AUTOMATIC SEGMENTATION ALGORITHMS

Patrick J. Mulroy

Applied Research & Technology,
BT Laboratories, Martlesham Heath, Ipswich, UK
<patrick.mulroy@bt-sys.bt.co.uk>

Abstract. Automatic video segmentation for the purpose of content definition remains an unsolved problem but it is now receiving widespread interest due to the development of the content based audiovisual coding standard ISO/MPEG-4. Several techniques aimed at both definition and tracking are being developed in the COST211ter simulation subgroup. This paper looks at implementation results of 4 automatic definition techniques - recursive shortest spanning tree, pyramid region growing, morphological watershed and colour clustering and compares these operating on the same video sequence test set.

1 INTRODUCTION

The COST211ter simulation subgroup is developing a common Analysis Model (AM), to evaluate video segmentation algorithms and to further develop the best. The group has produced the first version of the AM with the Recursive Shortest Spanning Tree (RSST) [2] technique used for both motion and texture segmentation. The general AM structure consists of parallel texture, motion and motion compensated texture segmentation stages combined in a rule based region processor to determine a final result. Over the course of the development of the AM it is hoped to compare the RSST algorithm in a controlled way with other automatic and semi-automatic techniques including morphology, change detection, clustering, region growing and fuzzy classification. This contribution presents a comparison of some of the automatic techniques for which implementations are currently available - RSST, pyramidal region growing, morphological watershed and colour clustering. The advantages and disadvantages of each technique for content extraction are then discussed.

2 ALGORITHMS

2.1 Recursive Shortest Spanning Tree

RSST is a relatively simple technique, recursively grouping neighbouring regions based on a link weight with a bias for merging small regions. In our case the link weight is colour (YUV value) difference and specifically for two adjacent regions, A and B, the distance between them is given by :

$$d_{AB} = \{(Y_A-Y_B)^2 + (U_A-U_B)^2 + (V_A-V_B)^2\}.\frac{sizeA.sizeB}{sizeA+sizeB}$$

where Y_X, U_X and V_X respectively represent the mean Y, U and V values of all pixels inside region X and sizeX is the number of pixels within the region. Figures 2(a)-2(d) show boundary images obtained using RSST on the COST211ter test data set. The technique can also be applied to motion vector fields using vector differences. The algorithm steps are :

```
• Initialisation
      No_of_Regions = Total number of pixels in image.
      Size of all = 1
• Loop While (No_of_Regions > Target)
      Calculate distance measure for all 4-connected
      region pairs
      Join two closest regions
      Calculate new region values and sizes
      Decrement No_of_Regions
```

2.2 Pyramidal Region Growing

This technique is an extension of [3] which uses a hierarchy of frames of increasing resolution and a competitive region growing process to produce a final segmentation. The basic method involves forming a truncated image pyramid, with each layer having a quarter of the nodes of the layer below. Each node has both a bottom-up (BU) and a top-down (TD) parameter associated with it and these are assigned in an iterative process. The BU parameter is an image measure (e.g. mean grey level) related to the associated 4x4 'son' nodes of the next larger layer. The TD parameter at the top layer is initially the BU parameter of the same node. For the next and subsequent layers the TD parameters are forced to take on one of the TD parameters of the nearest 2x2 'father' nodes on the upper (smaller) layer. The one with the closest BU parameter as the node under question is chosen. At the bottom layer the TD and BU values are made equal forming a partially segmented image and the process is repeated until stable. Where the pyramid is truncated determines the scale of the features extracted.

A number of problems with this technique have been addressed in [4] such as poor choice of region value, failure to segment 'slender' shaped regions and failure to concurrently segment regions of varying size. A mechanism to seed and erode new regions at different pyramid layers and an edge detection algorithm, to prevent arbitrary segmentation in areas of low contrast, have been incorporated. The parameters, SEED and ERODE, were adjusted for the best segmentation of the first frame of the sequence. The SEED threshold controls the number of new regions being seeded at each resolution whereas the ERODE threshold controls the amount of edge strength required to establish a new region. First frame results are given in figures 3(a)-3(d).

2.3 Watershed Morphology

Mathematical morphology is a well developed technique with considerable work reported in the literature on the application to image segmentation [5-8]. With the watershed approach a greyscale picture is considered to be a topographic surface with a pixel's grey level representing its elevation at that point. If we locate the regional minima of the surface, pierce holes at these points and then slowly immerse the surface into water we will progressively fill the different catchment basins of our surface starting from the minima of lowest altitude. Where the water coming from two different basins would merge we build a dam and we continue until the immersion is complete. We now have all our regional minima surrounded by dams, which mark out all our catchment basins, or regions, of the picture. A fast implementation of the watershed algorithm, based on [5], was used in this comparison. In this implementation, before the watershed algorithm is applied, several morphological filtering operations are performed to simplify the input image. These initial steps are :

- Filtering of the original image with the *morphological open-close by reconstruction operator* with a specified filter size. This operator removes regions smaller than a given size but preserves the remaining object contours.
- Calculation of gradient of filtered original by eroding/dilating using a 3x3 structuring window and calculating difference.
- Labelling of the flat regions of the gradient in a 'marker' image.
- Preservation of the highest peaks only in gradient between 'marker' regions

It is on this modified gradient that the core watershed segmentation is finally applied. To ensure that as many as possible of the required content edges were preserved a low morphological filter size of 5 was chosen for the test set. Results are shown in figures 4(a)-4(d).

2.4 Colour Clustering

Clustering of pixel feature vectors [9-11] is also a relatively simple technique where pixel property vectors (e.g. RGB or YUV) are iteratively reassigned to cluster bins based on a distance metric between individual vectors and cluster means. This is in effect a multi-dimensional histogram thresholding approach and is easily extendible to higher dimensions. The implementation available has been used successfully on texture segmentation problems with 15 component feature vectors.

Clustering based on a distance metric is very close to the RSST approach except here there is no connectivity constraint. This allows us to have disconnected regions with the same region identifier which may or may not be desired. Although figures 5(a)-5(d) show results that identify fewer than 8 cluster bins there are in fact many more disconnected regions as can be seen in the boundary images.

An extension on earlier clustering results reported has been the addition of morphological filtering techniques as a post-process. Specifically the open-close by reconstruction operator with a filter window of 3x3 has been used to clean up the raw clustering output.

3 RESULTS

3.1 Test Set

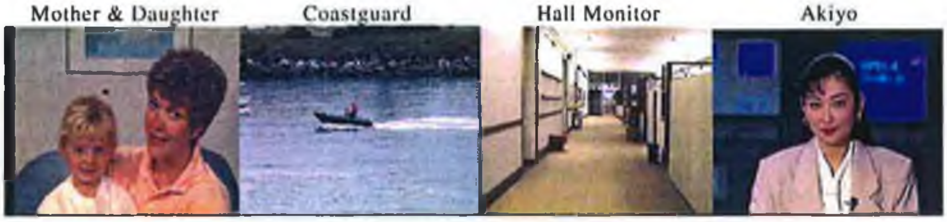


Fig 1(a-d) Original QCIF first frames

The test sequences used originate in the ISO/MPEG and ITU-T SG16 standardisation bodies. QCIF (176 pels x 144 lines x 30 fps) resolution sequences were used with 4:2:0 YUV colour format. For the techniques under test that use all the colour components, RSST and colour clustering, U and V are upsampled by pel repeat. First frame results of all techniques are shown below. Manual intervention was used to specify target and other parameters for the first frames. Results applied to each frame of the sequence have also been compiled and will be demonstrated to show temporal region stability. The stability issue is one that will be addressed by tracking algorithms and is not covered here.

3.2 RSST

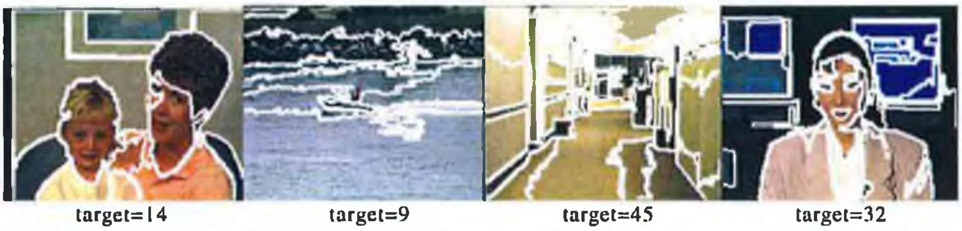
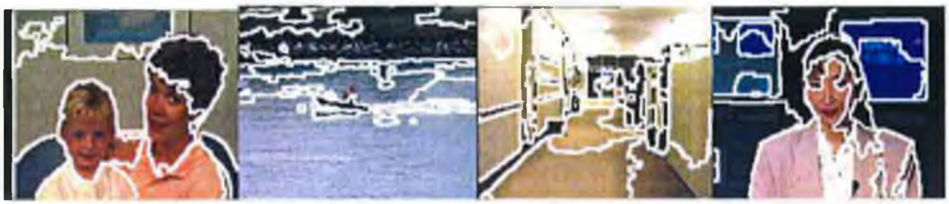


Fig 2(a-d) RSST Result, variable target number

This result shows what can be obtained with the current AM definition algorithm. Although some arbitrary region boundaries are shown, this is an encouraging result, with most important content boundaries detected. The algorithm is also the fastest tested on a Solaris 2.5, SPARCStation 20 platform taking an average of 3.2 secs/frame.

3.3 Pyramid Region Growing



S/E 3 / 5 S/E 1.1 / 5 S/E 1.1 / 5 S/E 1.3 / 0

Fig 3(a-d) Pyramid region growing result, variable seed/erode parameters

This technique performs well considering that processing is only carried out on the luminance component of the test sequences. It was the slowest algorithm tested taking an average of 13.5 secs/frame.

3.4 Watershed

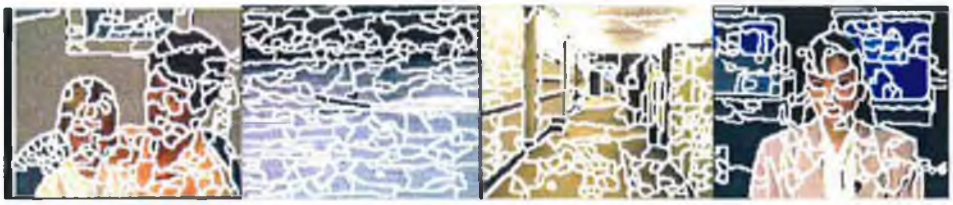


Fig 4(a-d) Morphological watershed, filter size of 5

The morphological watershed implementation was also fast with an average of 5.1 secs/frame. The results show many arbitrary boundaries but this is expected with such a small morphological filter size. At higher filter sizes there are fewer regions but also important content edges are removed. Also the arbitrary boundary regions could be merged in further processing, such as RSST, while retaining the good boundary edges.

3.5 Clustering

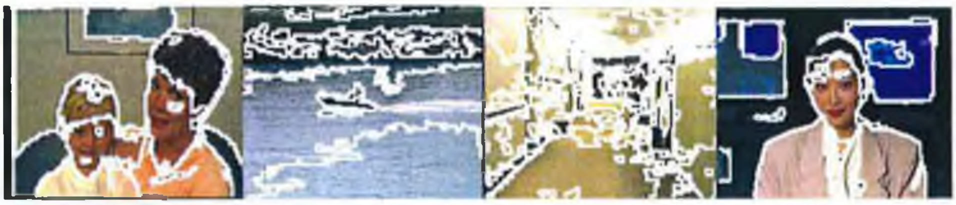


Fig 5(a-d) Colour clustering, target number of ≤ 8

Even without a connectivity constraint the clustering results show good connected content regions. Content edges are not as clean as other techniques and several arbitrary regions could be usefully reassigned. The algorithm also takes 11 secs/frame on average for processing due to its iterative algorithm. The *Hall Monitor* result did not converge with small number of 8 input clusters. Algorithm forces halt at more than 30 iterations.

4 CONCLUSIONS

The goal of this processing is to delineate true content edges while giving as few arbitrary boundaries as possible. With this criteria in mind the RSST technique performs well on the generic video test set with perhaps the pyramid region growing technique as the next best. In terms of complexity the RSST is also the fastest algorithm with the watershed next in speed. If the number of arbitrary watershed regions can be reduced while retaining all important edges this would be a good competitor for the RSST approach.

True content extraction from generic video requires a human perception of objects in the scene as opposed to a mathematical region model and any automatic algorithm is competing with $50\text{x}10^9$ neurons already programmed with a lifetime of experience in image analysis. We can only hope to design an automatic definition algorithm to make as few mistakes as humans do.

5 REFERENCES

[1] Koenen R, Pereira F, Chiariglione L, "MPEG-4 Context and Objectives:", <http://drogo.cselt.stet.it/ufv/leonardo/icjfiles/mpeg-4_si/paper1.htm>, Image Communications Journal Special Issue on MPEG-4.

[2] O. J. Morris, M.J. Lee, A.G. Constantinides "Graph Theory for Image Analysis : an approach based on the Shortest Spanning Tree", IEE Proceedings, vol. 1333, pp. 146-152, April 1986.

[3] Burt P J, Hong T-H, Rosenfeld A, "Segmentation And Estimation of Image Region Properties Through Co-operative Hierarchical Computation", IEEE Trans. Systems, Man, And Cybernetics December 1981, Vol 11, No 12, pp 802-809.

[4] Beaumont J M, "Image Segmentation based on a Neural Model", PhD Thesis, Imperial College of Science, Technology and Medicine, June 1996.

[5] Vincent L, Soille P, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol 13, No. 6, June 1991.

[6] Serra J, "Image Analysis and Mathematical Morphology. London: Academic, 1982.

[7] Sternberg S.R. "Grayscale morphology" Comput. Vision, Graphics, Image Processing, vol. 35, pp. 333-355, 1986.

[8] Maragos P, Schafer R.W., Butt M. A., "Mathematical Morphology and its applications to image and signal processing", Kluwer Academic, 1996.

[9] Mulroy P, "Spatial and Temporal Image Segmentation by Feature Clustering", Intl. Workshop on Coding Techniques for Very Low Bitrate Video, VLBV '95, Tokyo, Nov. 1995.

[10] Mulroy P, "Clustering as a Video Object Plane formation tool", COST211ter document SIM(96)04, Barcelona, February 1996.

[11] Ohm J.-R, Ma P., "Feature-based Cluster Segmentation of Image Sequences", COST211ter document SIM(96)43, Ankara, October 1996.

VRML GETS REAL THE MPEG-4 WAY

Patrick Mulroy

INTRODUCTION

Within the videoconferencing world, systems have existed for some time offering two way, and even multipoint, video, audio and data links. Further work on these systems is concentrated on attempts to convey more feeling of 'presence', for example to provide eye contact or to represent participants in a shared virtual space. In the virtual reality community, particularly within the Virtual Reality Markup Language (VRML) group, much work is in progress to define distributed virtual environments. These offer immersive environments, accessible through World Wide Web (WWW) browsers, where personal representations, or avatars, can move about and interact in 3D worlds, both with the world itself and with other avatars. The new VRML 2.0 specification, with Silicon Graphics *Moving Worlds* proposal [1], now enables much more dynamic and interactive environments; and software such as Dimension-X's *Liquid Reality* [2] help produce the Java code needed to give avatars and other virtual entities realistically complex behaviour.

The convergence of these technologies is now obvious, and desirable, and the upcoming MPEG-4 standards development will help precisely in this new domain. This paper details the current status of the MPEG-4 standard and discusses its possible application to the field of telepresence in shared virtual reality spaces.

OVERVIEW OF MPEG-4 STANDARD

MPEG, the Moving Picture Experts Group, is a working group of ISO/IEC, which has already produced two very successful digital video and audio coding standards: MPEG-1 addressed compression for storage applications at up to 1.5Mbit/s and MPEG-2 higher quality broadcast applications at 4Mbit/s and above. MPEG-4, the current activity, defines a multimedia applications standard spanning the domains of digital television, graphics animation and WWW content access and distribution. It is addressing audio and video compression, optimising over a much wider bitrate range than its predecessors, but it is also allowing coding of arbitrarily shaped video, coding of synthetic video and audio, integration of text and graphics and end-user interaction. The full standard set will be comprised of six parts. The schedule for progression is given below in table (i).

Part	Title	WD	CD	FCD	DIS	IS/TR
MPEG-4						
1	Systems		97/10	98/07	98/11	99/01
2	Visual		97/10	98/07	98/11	99/01
3	Audio		97/10	98/07	98/11	99/01
4	Conformance Testing	97/10	98/10	99/07	99/11	00/01
5	Reference Software - Technical Report (TR)		97/10	98/07	98/11	99/01
6	DSM-CC Multimedia Integration Framework (DMIF)	97/07	97/10	98/07	98/11	99/01

Table (i) : MPEG-4 Standards progression schedule.

Patrick Mulroy is with Applied Research and Technology, BT Laboratories, Martlesham Heath, Ipswich, UK.

Systems, Audio and Visual parts are the most advanced at this stage (May 1997) and will be introduced here. These parts are currently at working draft (WD) status and are expected to progress to full international standard (IS) status in January 1999. Audio (WD 14496-3) and Visual (WD 14496-2) will define a standardised coded representation of audio and visual content, both natural and synthetic, called "audio-visual objects" or AVOs. Systems (WD 14496-1), will standardise the composition of these objects together to form compound AVOs (e.g. an audiovisual scene), and multiplex and synchronise the data associated with individual objects, so that they can be transported over networks at appropriate quality of service levels.

MPEG-4 : SYSTEMS

Systems will be the core of the new standard set handling scene description, multiplexing, buffer management, synchronisation and content-related IPR identification. Scene description will be encapsulated in a new BInary Format for Scene description (BIFS) format. For the first Committee Draft (CD) the workplan of the systems group gives priority to the specification of 2D BIFS scene description. 3D capability may not be included in the first CD, even though it is illustrated in fig. (i), as implementations are not yet at a mature enough stage. If not included at this time this will be part of the systems extension which is due for completion after 1998. Also discussed as a candidate for the systems extension is an API definition of MPEG-4 algorithmic tools. This will support downloadable codecs and provide an unprecedented level of flexibility in an ISO multimedia coding standard.

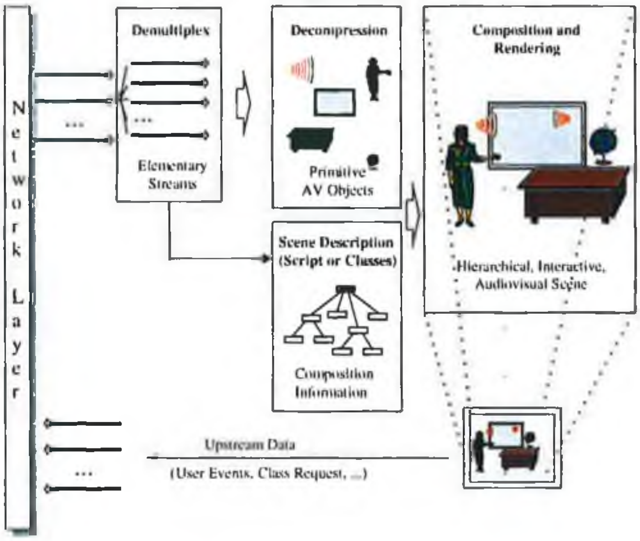

Network Layer
Demultiplex
Elementary Streams
Decompression
Primitive AV Objects
Composition and Rendering
Hierarchical, Interactive, Audiovisual Scene
Scene Description (Script or Classes)
Composition Information
Upstream Data
(User Events, Class Request, ...)

Figure (i) : Systems view of an MPEG-4 Receiver.
(Figure from MPEG-4 Overview [3])

Software projects in Java and C++ have already been launched to demonstrate the capabilities of the systems standard in July '97. They will develop and demonstrate a simple integrated systems layer, featuring 2D scene composition, synchronisation and multiplexing - all with real-time performance.

MPEG-4 : VISUAL

The visual standard will include all the coding tools relating to visual data, natural and synthetic, hence its name. Currently there are two Verification Model (VM) specifications: the video VM for natural video coding, frame based or arbitrarily shaped; and the Synthetic/Natural Hybrid Coding (SNHC) VM for synthetic 2D/3D graphics tools. As the tools are tested and proven within these VMs, they will be incorporated into the visual WD, and ultimately, if widely agreed on, in the IS.

For the highest efficiency in video compression the video VM is building on the work of the ITU-T Recommendation H.263, a recent standard optimised for low bitrate communication. Extensions have already been made to the core coder, most significantly the ability to code shape and transparency information of soalled Video Object Planes or VOPs, see figure (ii) below. The video coder will be capable of coding video t anything from 10 kbit/s to 4 Mbit/s and possibly higher.

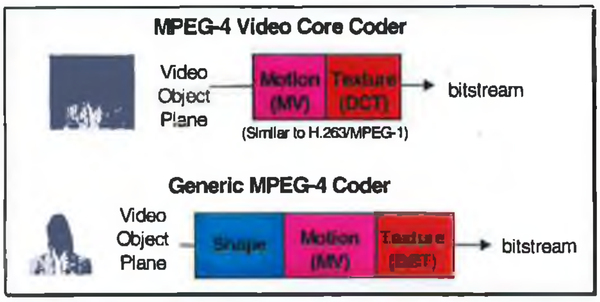


Figure (ii) : Video Verification Model stages

Transcoding between MPEG-4 and MPEG-1/H.263 elementary bitstreams should be straightforward in the rame based (or rectangular VOP) operating mode (e.g. when using the proposed real-time communications profile). The addition of arbitrary shape coding in the generic coder enables a range of new content-based unctionalities such as content manipulation in the compressed domain and content scalability. The shape coding is done using a bitmap based technique known as context-based arithmetic coding. This is similar to he binary image coding used in the ISO/JBIG standard and used in group 4 fax specifications.

The visual standard will not specify how shape and alpha (transparency information) planes are to be generated as this is a producer or encoder issue. MPEG philosophy has always been that only decoder issues should be specified to guarantee interworking. Competition is then between companies who can provide the best encoder engine, and also, in this instance, provide the most useful shape and alpha data. Blue-screen echniques, already in widespread use in the broadcast industry, are one source for this information. The video group is considering automatic video segmentation algorithms which may well form an informational annex to the standard to help content producers. Other automatic and semi-automatic segmentation techniques are also being investigated by the European COST211ter project and [4] gives a review of some of these.

The synthetic video components of the SNHC VM currently include media integration of text and graphics (MITG), face and body animation, texture coding (generic and view-dependent textures) and static and dynamic mesh coding with texture mapping. MITG provides the capability to overlay and scroll text, images and graphics on coded video backgrounds. Work on this is currently proceeding and is likely to be included soon in the visual WD.

Figure (iii) : Wireframe head model

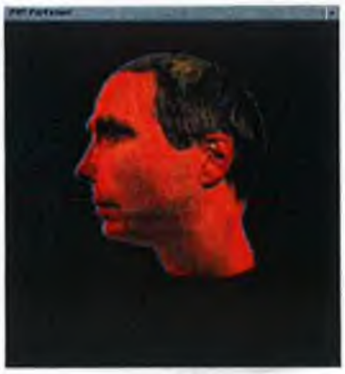


Figure (iv) : Texture mapped model

Face animation will allow definition and animation of synthetic 'talking heads' such as figure (iii). Only the face definition parameters (FDPs) and the face animation parameters (FAPs) need standardising here. Earlier work at BT Labs [5,6], combining this with texture mapping (figure (iv)), has shown how realistic these synthetic personae can actually be. Body animation issues have not been worked on yet but one proposal in this area known as *Jack* has already been input to the group.

MPEG-4 : AUDIO

The combined natural and synthetic approach of the visual standard is also reflected in the MPEG-4 audio work. Here compression of natural music and speech will be combined with structured descriptions of synthetic sounds.

The natural audio compression bitrate range is defined for MPEG-4 to be from 2 kbit/s to 64 kbit/s. A scalable coder using a combination of parametric, Code Excited Linear Prediction (CELP) and time to frequency (T/F) coding techniques will be used for this range. The systems layer will also provide for signalling use of other existing standards covering this range (e.g. ITU-T G.729 for speech coding). Above this range MPEG-2 Advanced Audio Coding (AAC) techniques will be used. AAC has been shown to significantly outperform all other audio coding standards in terms of bitrate and quality.

Synthetic sound currently includes Text-To-Speech (TTS) synthesis and score driven synthesis with MIDI and other control data as the score input. TTS coding will combine conventional text-to-speech, such as the BT Laureate[7] system, with prosodic parameters (pitch/inflection, volume, duration) that lend more natural intonation to speech reconstruction. Systems should be capable of coding prosodic-augmented speech as an audio object. Audio effects such as reverberation, spatialisation, mixing etc. will also be signalled in the systems layer to allow specialised decoders to apply effects locally to decoded audio, and merge and synchronise all audio objects before compositing with visual objects.

CONCLUSION

As a standards effort with wide industry participation, MPEG-4 will make a significant impact on the multimedia world. Where MPEG-1 and -2 have now helped to make digital television a reality, MPEG-4 hopes now to move multimedia communication from its current buzzword stage to being a real service on everyone's desktop. The integration with the VRML world will add another dimension to this work and should make provision of enhanced telepresence and virtual meeting room technology far more likely in the near term. In terms of the technological development in this area, the hooks into the VRML 2.0 specification are already being discussed jointly by the VRML and MPEG-4 Systems groups, and the core technology needed to develop the audiovisual coding and composition is already at an advanced stage.

REFERENCES

[1] Silicon Graphics, "Moving Worlds", http://vrml.sgi.com/moving-world
[2] Dimension-X, "Liquid Reality", http://www.dnx.com/lr
[3] Chiariglione L, "An overview of the MPEG-4 standard", http://www.cselt.stet.it/ufv/leonardo/paper/isce96.htm
[4] Mulroy P, "Video Content Extraction: Review of Current Automatic Segmentation Algorithms" to be presented at COST211ter Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '97), UCL, Leuven-la-Neuve, Belgium, June 1997
[5] Welsh W J, Searby S, Waite J B, "Model-based image coding", British Telecom Technical Journal, Vol.8 No.3, July 1990.
[6] Mortlock A, BT Talking Head Web Pages, http://www.labs.bt.com/showcase/head/index.htm
[7] Page J, Breen A, BT Laureate Text to Speech Synthesis Web Pages, http://www.labs.bt.com/innovate/speech/laureate/index.htm

Video coding — techniques, standards and applications

M W Whybray, D G Morrison and P J Mulroy

Virtually all teleconferencing systems incorporating video use compression algorithms to reduce the video data rate to manageable proportions. For a given bit rate, higher compression brings higher quality, although there can also be disadvantages, such as increased delay. Conferencing implies interconnectability, so the development of appropriate standards has played a vital part in enabling the videoconferencing market in particular to become established. This paper describes the main techniques used in video compression and how these have been embodied in successive generations of standards appropriate to various application areas. Other non-standardised techniques, and forthcoming standards are also discussed.

1. Introduction

Teleconferencing has developed through the stages of audio, and video and now data conferencing (including fax, file exchange, shared document editing, etc), and is poised to move on to virtual reality (VR) methods. VR may involve the mixing of real and synthetic video (for example placing the real video images of peoples' head and shoulders round a synthetic conference table), or even just purely synthetic video (for example using completely synthetic representations of people — 'avatars').

In the near future at least, a vital component for teleconferencing will continue to be the ability to *compress* the real video component, which commonly utilises by far the majority of the bandwidth. This paper reviews the fundamentals of video compression, the various standards that have emerged to cater for different application areas, and what the future might hold.

2. Video compression techniques

Analogue video connections for broadcast television consume in the region of 5 MHz bandwidth, but for teleconferencing purposes which require switching and long distance transmission, digital video is the only feasible approach. For digital operation, straightforward pulse code modulation of studio-quality video according to the sampling rates and quantisation law specified in ITU-R Recommendation BT.601 [1] requires a bit rate of 216 Mbit/s.

Clearly, when the above rates are compared to the 64 kbit/s of telephony circuits, the likely tariffs would stifle all but the most enthusiastic and wealthy customer's interest in videoconferencing and videotelephony services! Fortunately, once in a digital form video signals are amenable to compression.

Virtually all the compression algorithms in use today are waveform coders. These attempt to re-create, at the decoder, a replica of the original image waveform by using only the properties of a low-level representation. For example, an image is sampled to produce a set of picture elements, termed pixels, the brightness and colour of each being represented by numbers. Compression is achieved by exploiting properties both of these numbers and of the human visual system. No higher level understanding is undertaken.

An analogy is the difference between text transmitted by a facsimile machine and text transmitted as ASCII characters. The latter is at a much higher level and hence is a more efficient representation. Today, image compression is still the equivalent of the facsimile representation.

If the numbers are reproduced exactly, the coding scheme is said to be lossless and in this case is operating purely on statistical redundancy. Techniques which remove visual redundancy reconstruct images which differ objectively from the originals and are classified as lossy. The errors are either subjectively invisible or at an acceptably low level. There is no rigid definition of how much distortion is permissible because the acceptability is subjective, depend-ing on many factors, such as the picture material, the viewing conditions, the viewer's eyesight, the use to which the pictures are being put and, not least, the cost.

2.1 *Removal of statistical redundancy*

Statistical redundancy is not unique to image data processing. Many computer users, on finding that the capacity of their hard disk has become insufficient, have installed software which allows more data to be stored, or used file compression utilities to send files more efficiently by e-mail. Similarly most PSTN modems now incorporate data compression [2] which increases the apparent transmission speed.

Many of these techniques are based on the Ziv-Lempel algorithm [3] which recognises repetitions of parts of the data and sends a pointer to earlier occurrences of that data, rather than repeating the data in full. Depending on the characteristics of an image (natural or synthetic, background noise level, etc), the Ziv-Lempel algorithm may provide compression in the range just above unity up to three times.

However, the above algorithm does not make effective use of the fact that image pixels tend be highly correlated with their neighbours in space or time, which is the basis of many compression techniques that have been developed specifically for images and data from other natural sources (for example, audio).

Predictive coding

The basis of predictive coding is to use pixels or data already received to form an estimate or ‘prediction’ of the next pixels or data to be transmitted. A basic form of predictive coding is illustrated in Fig 1. Pixels are fed into the system in raster scanned order (i.e pixel by pixel scanned left to right across the image, line by line down the image, and picture by picture in a moving sequence). Differences are formed between each pixel and a nearby one temporarily stored in the delay element. The delay element is fixed and corresponds to selecting the pixel immediately to the left, the pixel immediately above (both intraframe predictions) or, in motion video, the pixel in the same place in the previous picture (interframe prediction).

The difference or ‘prediction error’ is seldom exactly zero because even in regions with no detail or in stationary areas the camera system usually adds some noise. However, the variance of the prediction error is generally substantially less than that of the original and further techniques such as entropy coding (see later in this section) can be used to represent the data with fewer bits than the original pixels. The original pixels can be reconstructed from the prediction errors by a decoder which consists of just the components in the outlined box.

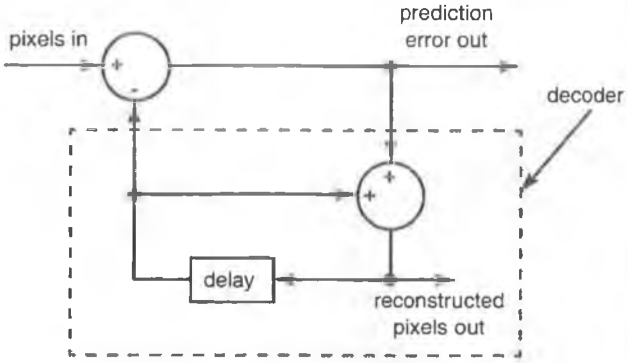


Fig 1 Basic predictive coder.

As well as using just one previous pixel as the predictor more complex schemes employing several previous pixel together may be used, for example a local average of a fev spatially adjacent pixels may provide a better estimate o the new one by averaging out noise, or estimating anc extrapolating the local slope of the image function.

When predicting from one picture to the next in movin video, the single pixel predictor is of course very good ir regions of the image where there is no change (or ‘motion’) but often very bad where there is actually motion Unfortunately the latter is precisely the area where mos data needs to be sent and compression is most needed! Thi observation led to the introduction of conditiona replenishment, whereby in areas of the picture where there is no temporal change at all, it is sufficient for a decoder tc simply display the pixels from the previous frame again. Ir changed areas, new pixels are transmitted using predictior from spatially adjacent pixels. Omitting the redundant transmission of unchanged pixels gives a big gain ir compression.

To achieve higher compression in the changed areas, motion compensated prediction was developed.

Motion compensation

This technique is an extension of the above prediction method, in which the delay for each pixel is varied dynamically throughout each still image to find the place in the previous still which is the best prediction. The corresponding horizontal and vertical offsets to refer to that best prediction are included in the coded bit stream. Ideally these offsets would be sent for every pixel but the overhead of doing this is much more than the savings obtained from the smaller prediction error. Instead, a single pair of offsets is applied to many pixels, a 16 × 16 block of them being a common compromise.

Motion-compensated prediction works fairly well on videoconference type scenes where the gross movements of people’s heads and bodies are represented to a first pproximation by simple translations of their positions from he previous image. However, real world objects also ındergo rotation, deformation, occlusion and so on, which re not well modelled by the above block-based motion ompensation. More complex schemes have been devised to nodel these higher order changes in image structure, ncluding representing the motion by complex warping unctions, and delineating the boundaries between areas vith different motions.

Entropy coding

Entropy coding is a general term for lossless data compression methods which rely on the statistics of a set of events' to be compressed. In most practical cases this neans the overall frequency of occurrence of the various events in a set. The length of the codeword used to convey a particular event is matched to the likelihood of it occurring. Shorter codes are used for the frequent events and longer codes for those appearing less often, hence the term variable length coding'. For example, in Morse code a single dot represents the frequently occurring letter 'e' whereas the rarer letter 'q' is encoded as dash, dash, dot, dash.

It can be shown [4] that the optimum length of a codeword for an event of probability p is $\log_2(1/p)$. If all possible events in the set are assigned codewords according to this formula, the overall bit rate required to send events from that set with the given probabilities will be minimised. Thus for an event with a probability of occurrence of 1/8 the optimum codeword length is 3 bits. Unfortunately in practical cases the formula will yield non-integer values for codeword lengths, and a means is needed to optimally assign integer length codewords to events. Huffman [5] devised an algorithm to do this, hence the term Huffman coding (which is often wrongly used as a term for variable length coding, which need not, in general, be optimal).

Huffman's method assigns integer length codewords to events, but this results in a loss of efficiency since the theoretically optimal codeword lengths are non-integer. Arithmetic coding [6] is a technique which overcomes this by not having a one to one mapping of events to codewords, and thus comes closer to the optimum compression. However, it is also more difficult to resynchronise the decoder in the presence of errors.

Entropy coding gives very little compression if applied directly to image signals because the distribution of the brightness levels is fairly uniform. However, prediction errors have a very peaked distribution centred about zero and variable length coding is very worthwhile.

2.2 *Removal of perceptual redundancy*

The degree of compression obtainable from lossless techniques is modest, typically rather less than 5:1. To achieve lower bit rates, it is necessary to employ lossy compression methods. Most complete algorithms incorporate both types.

Subsampling

A straightforward approach is to reduce the number of samples in the originals. After decoding, the missing samples can be replaced by repeating, or, better still, interpolating from, those that were coded. This lowers the spatial or temporal resolution but may not materially affect the usability of the resulting pictures. Unfortunately, the coded bit rate does not decrease linearly with the sample rate because the fewer samples have less correlation and are less amenable to compression.

Colour images are usually represented by three colour components. Red, green and blue are used for image capture and display purposes, but for transmission and compression an alternative format known as YUV is preferred. RGB can be converted to YUV and back by a linear matrix operation. The Y or luminance component represents the brightness, whereas U and V are 'colour difference' signals, which both have zero value in black, grey and white parts of the image, and are non-zero where colour is present[1]. Because the human eye has a higher visual acuity for the luminance component of images compared to the colour component it is possible to reduce the spatial resolution of the U and V signals significantly compared to the Y component with no perceptible loss of quality. In videoconference applications this has led to the use of image formats such as common intermediate format (CIF) where the Y resolution is 352 by 288 pixels, and U and V are reduced to 176 by 144 pixels each.

Quantisation

Quantisation reduces the number of discrete values a variable can take, reducing the number of bits needed to encode it. Accuracy is lost — a number of different input values will be the same at the output. Recommendation BT.601 allocates 8 bits each to the Y, U and V components, this being near the minimum necessary to avoid a 'painting by numbers' effect called contouring (although greater numbers of bits are often used during editing and manipulation stages to ensure adequate resolution is retained in the final result). There is not much scope for further quantising the original pictures directly without introducing visible degradation.

This barrier does not apply to prediction errors, which are generally largest at edges or fine detail in the originals. The human visual system is much less sensitive to errors in high detail regions — a phenomenon called spatial masking. This can be exploited by applying a nonlinear quantiser to

[1] To be more precise, Y, U and V are usually represented as 8-bit numbers, which are interpreted as the range 0 to 255. U and V take the mid-point value of 128 in black/grey/white regions, and fluctuate above and below 128 in colour regions, so 128 is their notional 'zero' value.

the prediction error so that large errors are encoded less accurately than small ones.

Transform coding

Transform coding is a mathematical operation which transforms a set of numbers to another set with more favourable properties, while retaining the same information. The discrete cosine transform (DCT) is one of the more popular transforms for image compression.

The DCT is usually applied to a block, typically 8 × 8, of original image pixels or prediction errors, and produces an equal number of coefficients which are in many ways similar to Fourier transform coefficients. They contain, in order, increasingly higher frequency components of the original data.

For natural images, these DCT coefficients have the useful property that over a large ensemble of blocks, the probability distributions of the coefficient values are highly peaked around the value zero, and can be efficiently compressed using entropy coding. Additionally, the variance of the probability distributions decreases with increasing spatial frequency of the coefficients such that after quantisation, many of the coefficients are zero and can be discarded. This is illustrated in Fig 2, where a complex dark to light shading across an 8 × 8 pixel block results in only a few DCT coefficients having significant non-zero values. Furthermore, the eye is less sensitive to the higher spatial frequencies allowing them to be more coarsely quantised, resulting in additional compression.

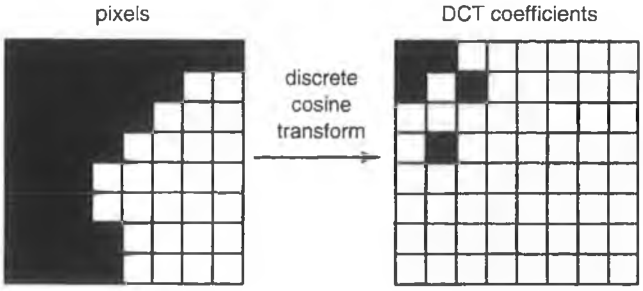


Fig 2 Discrete cosine transform of an 8 × 8 block of pixels.

Wavelets

Wavelets [7] are another way of transforming an imag into an alternative representation based on spatia frequencies, in a similar manner to the DCT. Howeve whereas the DCT is performed on a block of pixels at a tim to produce a block of DCT coefficients representing th spatial frequencies within the block, wavelets are usuall used on the complete image to split it into two spatia frequency bands in each direction (usually known as the I or low and H or high bands), yielding a total of four sub bands.

These four bands are designated LL, LH, HL, and HH The LL band is essentially the original image reduced i scale by a factor of two in both directions, whereas the othe bands represent higher frequency details associated wit horizontal, vertical and diagonal edges or other detail. Th LL band is usually subject to one or more further rounds o wavelet decomposition, resulting in a hierarchy of bands a illustrated in Fig 3.

As with the DCT, the wavelet transform itself gives n compression directly, but it maps the image into a domai where the significant data is structured in a way which i amenable to compression. Thus, apart from the LL band, th resultant wavelet coefficients have highly peake distributions around zero (as with the DCT coefficients) and entropy coding can be used. Secondly, the locations o non-zero coefficients in the various bands are usuall spatially aligned and quadtree type addressing schemes ca efficiently indicate those coefficients to be transmitted wit low overhead [8]. Thirdly, the hierarchy of spatial frequencies provided by the wavelet transform are a goo match on to the spatial frequency sensitivity of the huma eye, allowing coarser quantisation to be used on the highe frequencies.

Because the wavelet transform operates equally on all pixels in the image rather than grouping them arbitrarily into blocks, the coding distortions produced by excessive quantisation tend to be less visible and objectionable than those produced by block transform coders, where the block structure itself often becomes visible at high compression.

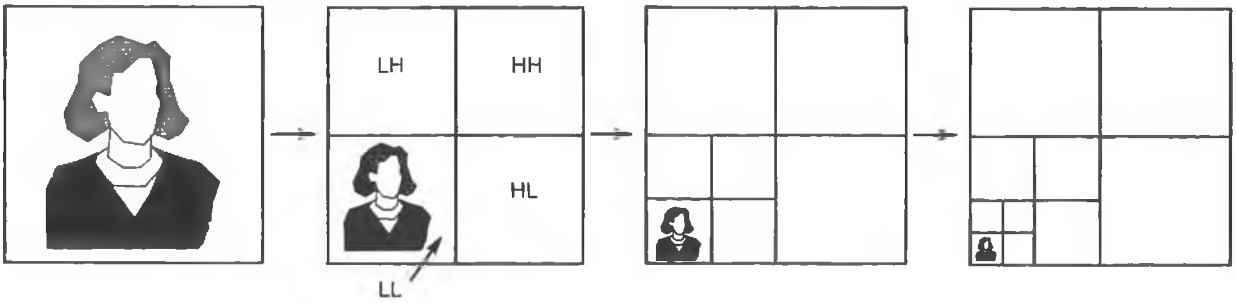


Fig 3 Progressive stages of wavelet transform of an image.

Vector quantisation

Wavelets and the DCT exploit the correlation of adjacent pixels by generating coefficients that have small or zero value if adjacent pixels are similar. Another way to exploit this correlation is vector quantisation (VQ) [9]. A block of pixels or prediction errors is quantised as one unit (a vector), rather than a pixel at a time. Typically, 4 × 4 sized blocks might be used, and a codebook of hundreds or thousands of possible example block patterns would be searched to find the pattern giving the best match. Only the index number of that entry in the codebook need then be transmitted.

VQ has a simple decode operation, being merely a table look-up, but the encoding process can be very demanding because at worst each block will require a full search of the codebook. In practice, various ways of pre-normalising the data, and structuring the codebooks can reduce this problem. VQ can also be used to quantise the data resulting from other processes such as DCT or wavelet transformation, and both theoretically and practically should perform better than an equivalent number of independent scalar quantisers. However, there are some problems concerning construction of codebooks to adequately cover the range of vectors to be quantised, and of the processing power required to perform the codebook searching at the encoder.

Fractal image coding

Fractal coding is another way of exploiting redundancy in images. It is similar to VQ except that no explicit codebook is required. It relies on the fact that in any given image it is usually possible to find one part of the image which, when suitably scaled down in size, rotated and grey-scale adjusted, provides a very close approximation to another part [10, 11]. By dividing an image into small regions (usually square blocks) and finding the appropriate 'mapping function' for each region, the whole image can be represented purely as a set of mapping functions. The surprising thing is that as long as these mapping functions are contractive (meaning essentially that the size and grey level must always be scaled down in the mapping) no actual image data at all is required to initialise the process. By starting with any image and applying the mapping functions repeatedly, the resulting image will finally converge to one close to the original image used to select the mappings. Depending on the scalings this would typically take less than ten iterations of each region mapping to converge to the endpoint.

Fractal coding can achieve very high compression in specially selected cases, but for typical images it is comparable to other methods. However, it has some other useful properties:

- it retains edge sharpness well,
- on natural images it can generate textures that although not faithful to the original image are quite often subjectively acceptable,
- the mappings found during compression can be used to scale images up or down in pixel resolution — although scaling up cannot accurately regenerate information previously discarded, the results can be subjectively pleasing,
- the decoding process is simple.

A major drawback is that the encoding process is very computationally intensive and to achieve real-time encoding it is currently necessary to compromise the compression efficiency by simplifying the process.

3. Standards and applications

The preceding section has highlighted most of the commonly used image compression techniques. This section reviews the historical development of image compression standards.

After some proprietary videoconferencing codecs operating at 6 Mbit/s from US and Japan, the European COST 211 project developed by 1984 a 2 Mbit/s codec [12] which was subsequently adopted by the CCITT as Recommendation H.120. This was based on conditional replenishment, changed areas being updated using an intraframe pixel prediction method.

A few years later more sophisticated techniques were applied in the development of ITU-T Recommendation H.261 [13]. This provides the motion video component of ITU-T Recommendation H.320 [14] for videophone and videoconferencing services at total bit rates between 64 and 1920 kbit/s, and embodies the basic principles from which subsequent algorithms including MPEG 1 and 2, and H.263 were developed through a process of progressive refinement and addition of new features.

3.1 H.261

The picture format used for H.261 is either CIF (see section 2.2) or Quarter CIF (QCIF) where the numbers of pixels in each dimension are half those of CIF. Remembering that the colour components U and V are at half the spatial resolution of the Y pixels, Y, U and V pixels are grouped separately into 8 × 8 blocks, and then four Y blocks and the spatially corresponding U and V blocks are associated into a 'macroblock' as shown in Fig 4.

The structure of an H.261 encoder is shown in Fig 5 and can be seen to combine several of the techniques discussed in section 2.

The encoder works as a predictive coding loop by analogy with Fig 1. It uses past data to form a prediction of

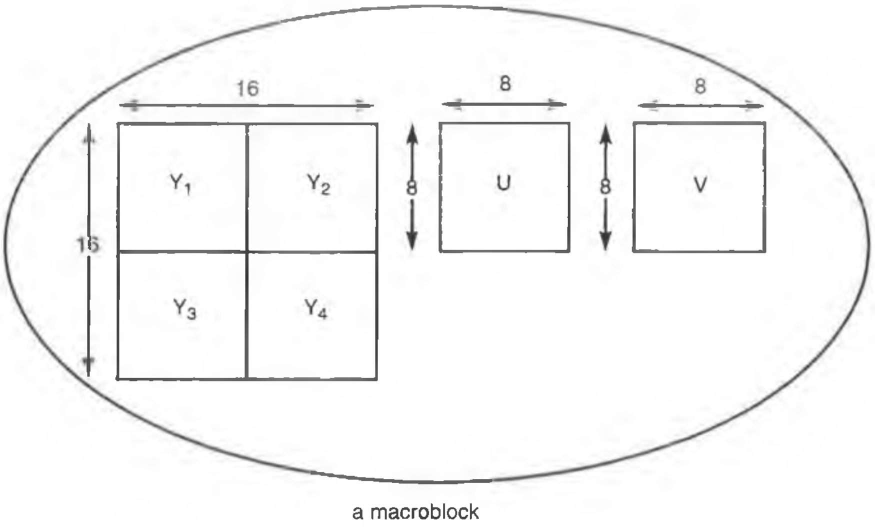


Fig 4 A macroblock composed of four Y blocks and a U and a V block

the current input data and quantises the prediction error for transmission. It then performs the 'decoder' functions of inverse quantisation and summation with the prediction to reconstruct an approximation to the input data.

To understand the coding loop in more detail, begin with the picture store, which holds the previously decoded picture. The motion estimation unit compares the incoming picture, a macroblock at a time, with this stored picture to determine the best motion vector to use for each macroblock. Having done so, it directs the picture store to output pixels (an 8 × 8 block at a time) to the rest of the coding loop, using the previously calculated motion vectors to select the appropriate parts of the stored picture. When the switch is in the 'inter' position, a block passes through a low-pass spatial filter which has been found to improve coding performance, and is then used as the prediction for the corresponding input block. After subtraction to yield the prediction error, the block undergoes a discrete cosine transform, the DCT coefficients are quantised, and the non-zero coefficients addressed and passed to the variable length coder. An inverse quantiser generates the appropriate reconstruction levels for the quantised coefficients and passes them through the inverse DCT to give an approximation to the prediction error, with some added noise due to the quantisation process. On adding this to the prediction, the reconstructed picture is produced, which is close to the input picture, but with the added quantisation noise. As this noise is generated in the transform domain, it appears in the image as a fairly evenly distributed random component,

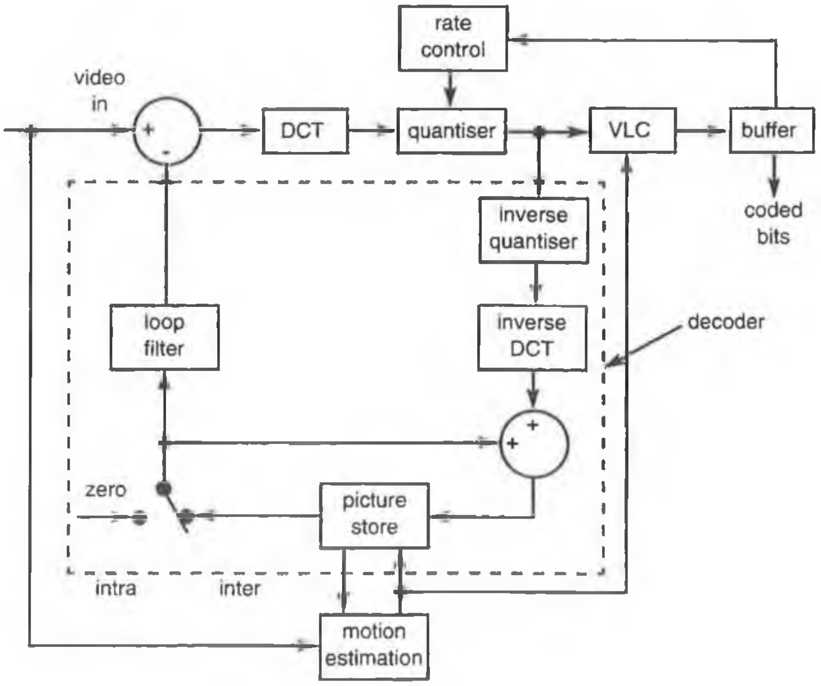


Fig 5 H.261 encoder.

though, at high compression, blocking and ringing artefacts are also visible.

The encoder is completed by a buffer which receives data generated by the coding loop at a variable rate and outputs it at a usually constant channel rate, and a rate control mechanism which adjusts the quantiser and also usually the coded picture rate to keep the long-term output bit rate the same as the channel rate, and the short-term variation within the range of variation that the buffer can absorb.

The H.261 decoder consists of the components within the outlined box in Fig 5, preceded by an input buffer and variable length decoder. As the H.261 decoder does not have to perform the motion estimation function or the forward DCT, it requires much less processing power than the encoder.

The prediction generated within an encoder must match that generated by a decoder, or else the contents of the two respective frame stores will progressively diverge as more frames are coded. Ideally the 'decoder' part of the encoder loop would be defined to be mathematically exactly equivalent to all decoders. However, in practice there are several ways of implementing the DCT which, because of the effects of rounding errors, can result in small differences in the numbers produced in different systems. Although the magnitude of this error is specified and constrained by the H.261 specification of the DCT accuracy, it can still accumulate and cause encoder/decoder mismatch. As a means of constraining this divergence, in addition to the predictive- or inter-coding mode, the encoder can also use the switch in Fig 5 to set the prediction to zero for a macroblock or a complete picture, in which case the input signal is coded directly in 'intra' mode. Since this mode does not rely on previously transmitted data it is used to initialise the encoder and decoder loops to the same state at the start of a session, to clear any transmission errors that may have resulted in the encoder and decoder loops getting out of step, and to control long-term build-up of rounding errors in the DCT. Intra mode is not used more often than necessary as it does not provide as much compression as inter mode.

The H.261 coder is only formally specified to work on CIF and QCIF pictures, and at bit rates in the range 64 kbit/s up to 1920 kbit/s. Although it can be used outside this range, other codecs such as MPEG 1 and 2 are generally more suitable at higher rates, and H.263 at lower rates. The main application of H.261 is still videoconferencing using the H.320 Recommendation. On a 2B channel ISDN connection the speech is usually coded with G.728 at 16 kbit/s leaving 108.8 kbit/s for the video (after deducting multiplexing overheads), at which rate H.261 provides adequate quality to cope with for two or three people at each terminal with moderate amounts of motion. For higher numbers of people per terminal, or for longer conferences, a higher bit rate provides higher quality and less user strain. H.261 is now also being used in other areas including streaming of video from web sites, and videoconferencing using local area networks (H.323).

3.2 MPEG-1

The H.261 coding methods were further refined in ISO MPEG-1 [15] and MPEG-2 [16], which are aimed at a wider range of applications including those where storage is an important aspect.

MPEG-1 added the technique of half-pixel motion estimation to H.261, which used only integer pixel estimation. This allowed motions to be more finely tracked and improved the prediction and hence compression performance. The second main change was to add the concept of 'B-Pictures'. These are pictures in which predictions can be formed from a picture occurring later in time than the current picture, instead of just from the previous picture as was the case with H.261. Using such bi-directional predictions (hence B-Pictures) improves the overall performance. The ability to look ahead in time is accomplished by storing up a short set of pictures in frame buffers, coding the latest stored picture in the normal forward prediction (P-Picture) mode, then going back and coding the remaining stored ones in B-picture mode (Fig 6). Although B-Pictures offer significantly higher compression than P-Pictures (by a factor of typically 2 to 3), which in turn have higher compression than I -Pictures (by a similar factor), there is a penalty of increased picture delay, and more complex processing requirements. The delay of several pictures (equivalent to typically 100 ms) is usually only a problem for real time conversational services such as tele-conferencing, for which the delay can be minimised at the expense of compression by using few or no B-Pictures.

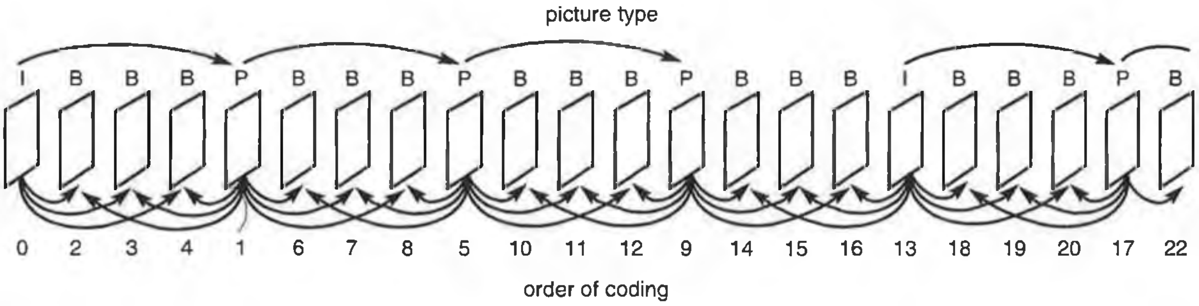


Fig 6 Order of coding and prediction paths of I, P and B-pictures.

The compression achieved by MPEG-1 is around 30% greater than that of H.261, though this is somewhat dependent on picture material and bit rate.

Figure 6 also illustrates the regular insertion of intra-coded pictures (I-Pictures) in a typical MPEG sequence. Since these can be decoded independently of any preceding or following frames they are used to ensure rapid recovery from any transmissions errors, and allow random access (i.e. jumping into the compressed video stream at any point). They also provide a means of playing the video sequence backwards as is necessary to duplicate video cassette recorder functionality such as fast reverse play — essential for video-on-demand type applications and for general browsing of compressed video. MPEG-1 was originally developed for use at single-speed CD ROM bit rates of around 1.2 Mbit/s and CIF resolution. It is, however, suitable for other applications not requiring interlaced pictures (see section 3.3), such as the interactive multimedia services trials run by BT [17] where a video bit rate of 1.6 Mbit/s was used to deliver roughly VHS quality video to homes for a mix of entertainment, education and information.

3.3 MPEG-2

MPEG-2 was developed in collaboration with ITU-T and is also known as ITU-T Recommendation H.262. It extended MPEG-1 to deal with the interlaced scanning of conventional television systems. Interlace is a hangover from analogue transmission and display systems which addressed the problem that for a good rendition of motion only 25 to 30 pictures per second are required, whereas at least 50 screen refreshes per second are required to avoid annoying flickering. In cinema this is done by displaying each picture several times. In analogue television there was no means of storing a picture at the receiver for repeated display, so each picture or 'frame' was scanned as two 'fields', each using half the number of lines in the frame, with the lines spatially interlaced to give double the number of lines for the frame, effectively increasing the vertical resolution. So 50 (or, in the USA-developed NTSC system, 60) fields per second are delivered to reduce screen flicker, at 25 (or 30) frames per second. When digitally coding such pictures it is important to realise that the two fields were captured at different times, and thus any moving object appears at different positions in them. If the fields were coded as a single frame there would be some very complex high-spatial frequencies to deal with as the alternate lines of the fields contain uncorrelated information. MPEG-2 deals with this by having extra prediction modes which can predict from one field to another in appropriate ways to extract the useful correlations remaining. By coding interlaced TV signals, MPEG-2 can be used to deliver broadcast (BT.601) resolution, at around 4 to 8 Mbit/s. MPEG-2 can also be used to encode higher resolutions such as HDTV, which was to be the subject of a separate 'MPEG-3' phase, but was subsumed into MPEG-2.

As with all video coding schemes, the quality achieved depends on the complexity and motion in the pictures, and on the available bit rate. Since historically the networks available have operated at fixed bit rates it has been usual to operate video compression schemes with a buffer and control system to meet that fixed bit rate. However, the result is that the video quality actually varies through the sequence — dropping in the more complex parts and perhaps using more bits than is necessary for acceptable quality in others.

With the arrival of networks that can support variable bit rate transmission such as ATM, it should be possible to operate with constant picture quality by allowing the bit rate to vary as necessary. Similar ideas can be used on satellite or cable delivery systems where a total bit rate budget can be shared adaptively between several video channels according to their relative need, achieving a higher overall quality than a fixed bit rate budget for each. Thus in reality it is difficult to put a single figure on the bit rate required for 'broadcast quality video', and the 4 Mbit/s mentioned above would be adequate for most purposes, but will not prevent some visible impairments on complex scenes such as sports with rapid camera and player motions.

Another new feature introduced by MPEG-2 is 'scalability', whereby one video bit stream can be used to bootstrap the quality of pictures provided by a second video bit stream, and so on. An alternative view is that one compressed stream contains within itself sub-streams which can also be decoded to give lower quality pictures than the full stream. This scalability involves using the decoded pictures of the lower bit rate stream as sources of prediction for the next layer, which can be configured to provide a higher spatial resolution, a higher temporal resolution (more pictures/second), higher quality (less distortions), or a combination of these. A drawback is that the quality achieved by using all the layers is usually lower than that achieved in a single layered encoding at the same bit rate.

There are several applications for scalability. One is where a network may offer several levels of service such that a lower bit rate stream is guaranteed timely arrival, whereas one or more higher rate streams may suffer some data loss. The lowest layer guarantees that a picture is always available for display, while the other layers enhance the quality in a way that is resilient to occasional data loss. This scenario can apply in ATM and in Internet protocol (IP) networks.

A second application is multicasting on IP networks, where a single server sends out the multiple layers, and clients receive as many layers as network conditions allow them to, adding and dropping layers dynamically as network congestion changes, to receive the best quality video possible at any one time.

3.4 H.263

With the advent of faster modems, especially V.34 yielding 28.8 kbit/s [18], V.34 extensions to 33.6 kbit/s (full duplex) and 56 kbit/s (though not bi-directional at this rate), the possibility of providing useful motion video over the PSTN has emerged. The experience gained from H.261 and MPEG has been further augmented by development of the H.263 video compression algorithm [19]. The main aim for this algorithm was higher compression, given the limited bit rate available on the PSTN, yet low delay as the main application was conversational services. This aim was largely achieved as it delivers around twice the compression of H.261, though, as with the gain provided by MPEG-1, this is dependent on picture material and bit rate.

A significant departure for H.263 was to make some of the coding enhancements optional, to be negotiated between an encoder and a decoder. These optional modes appear as annexes to H.263.

- Annex D — unrestricted motion vector mode

  This allows motion vectors to point outside the normal picture boundaries by extrapolating out the edge pixels, and is particularly useful if the camera or whole scene is moving. This mode also doubles the allowed range of motion vectors from 16 to 32 pixels.

- Annex E — syntax-based arithmetic coding mode

  This substitutes arithmetic coding for the usual variable length coding procedure, and achieves slightly higher compression efficiency.

- Annex F — advanced prediction mode

  This allows motion compensation on 8 by 8 blocks as well as the usual 16 by 16 ones, and also uses a technique called overlapped block motion compensation, whereby a degree of overlap of adjacent compensated blocks is used to smooth out some of the block edges otherwise generated by block-based motion compensation.

- Annex G — PB-frames mode

  This allows the use of bi-directional prediction (see section 3.2) for increased compression. In contrast to the MPEG B-Pictures, here a P and a B picture are combined together into one data structure.

Figures for the amount of extra compression achieved by and the subjective effect of each Annex on different types of sequence are given in Whybray and Ellis [20]. Figure 7 shows a comparison between H.261, the base level of H.263, and H.263 with all four optional annexes (as described above) turned on.

The intended application for H.263 was the H.324 PSTN videophone Recommendation, which was ratified by the ITU in 1996, with products appearing in early 1997. Many of these H.324-compliant videophones were implemented as PC applications, requiring only a suitable modem and camera interface card to convert a multimedia PC into a videophone, with the video coding algorithm implemented in software on the PC. H.263 has also found favour with organisations providing streamed video on the Internet, where again the high compression efficiency is a benefit.

Further enhancements to H.263 (the main ones being listed below) are under development in the ITU in draft form at the time of writing, and are due for formal ratification (Decision, in ITU terms) in January 1998.

- New picture types:

  — scalability — use of enhancement layers of H.263 coded video to build higher resolutions/qualities at high bit rates in an embedded manner,

  — custom source format — enables coding of pictures with arbitrary frame sizes, and arbitrary pixel aspect ratios,

  — improved PB frames — improved version of an existing picture type for higher compression.

- New coding modes:

  — advanced intra coding — improved compression of intra frames,

  — deblocking filter — reduces blocking artefacts to improve subjective quality,

  — slice structure — groups macroblocks for improved error resilience,

  — reference picture selection — speeds up recovery from channel errors by use of a back channel,

  — reference picture resampling — improves prediction process by allowing affine transformation of prediction picture,

  — reduced resolution update — helps maintain a high frame rate where there is high motion,

  — independent segment decoding — improves error resilience,

  — alternate inter VLC — slightly improves inter compression,

  — modified quantisation — allows more rapid control of the quantiser, and increases quantiser range.

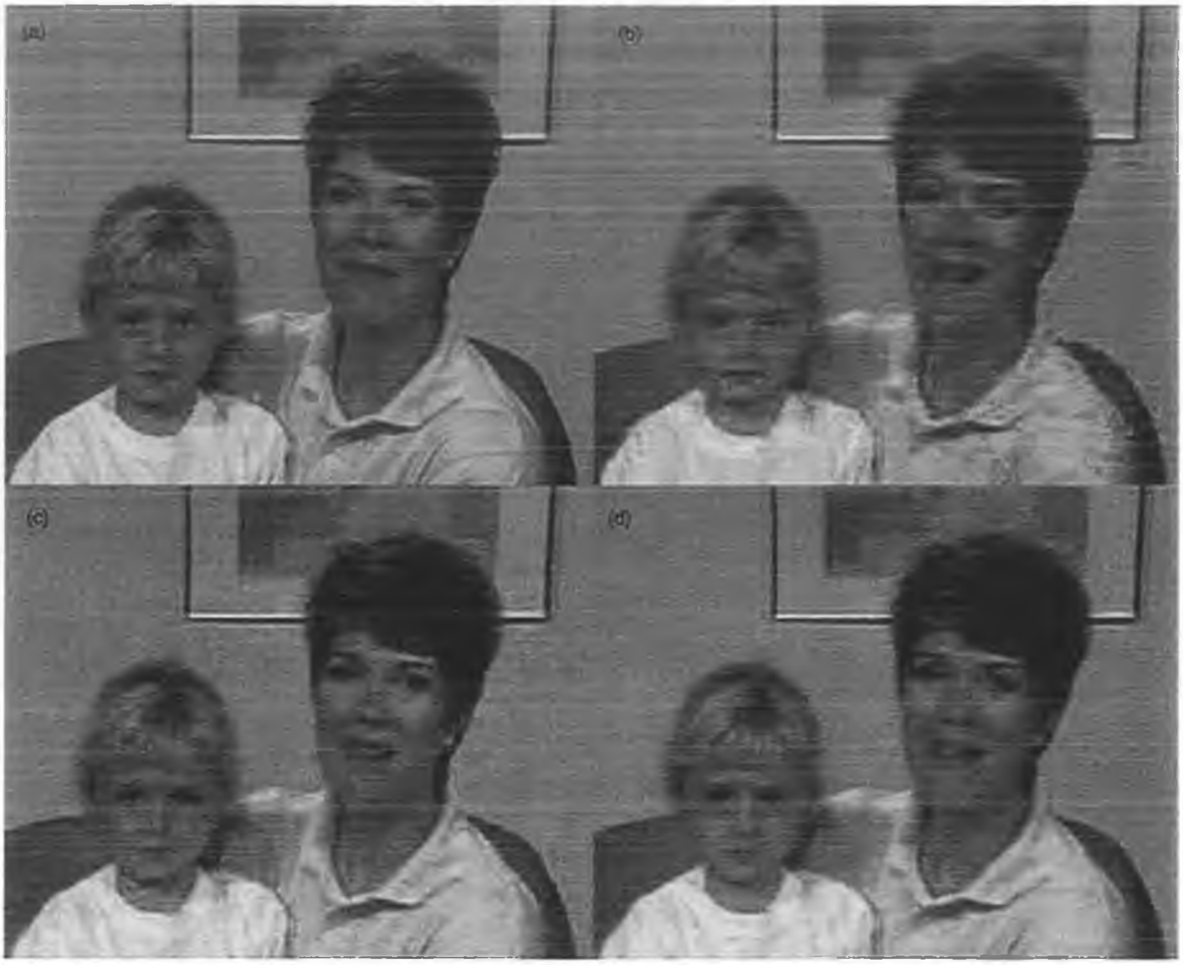


Fig 7 Stills from sequences coded at 28.8 kbit/s, 12.5 pictures/second. The four pictures are: (a) original frame (b) H.261 (c) H.263 base level and (d) H.263 with options D, E, F and G on [20].

The large number of these optional modes extends the application areas for H.263, but has caused concerns about compatibility. The existing scenario in conversational services is that the end-points should negotiate (normally using ITU Recommendation H.245) which modes are common and can be used, though the complexity of this negotiation may be getting out of hand. For other applications where there is limited scope for negotiation, for example streaming of pre-compressed video from a server, the decoder must be able to handle all modes used by the encoder.

3.5 JPEG

Regarding still pictures, in the 1970s, several telcos had introduced or were considering videotext services. Using similar formats to teletext, these were character based, providing text or block mosaic graphics. There was a desire to add 'photographic quality' pictures but a full screen image at Recommendation BT.601 quality is more than 800 Kbyte. At the then typical modem rate of 1200 baud, such a still would have taken about 100 minutes to receive. Even with smaller images and some relaxation of their quality, the waiting times would have been unacceptable. With photographic videotext as a catalyst, the usefulness of a generic compression standard for still images was widely appreciated. As a result the ISO ahd ITU-T jointly developed the JPEG algorithms [21, 22] which are equally suited to both storage and transmission applications. The basic JPEG algorithm is very similar to the intra-frame coding modes of the moving picture standards, being based on an 8 × 8 DCT. In teleconferencing, JPEG will often be used as the compression method for sending still images between conference participants as part of some higher level teleconferencing application.

4. Video transcoding

Compression algorithms are normally used to compress video to a known target bit rate, either for immediate transmission at that bit rate, or for storage and later playback. Once the compressed bit stream has been produced, it is usually not possible to directly access and decode it at a different bit rate. However, in the future it will increasingly be necessary to do this to support services such as:

- video databases which can be accessed at multiple bit rates,

- interworking adapters between networks carrying video at different rates, for example PSTN/ISDN,
- continuous presence multipoint videoconference units, where the input and output bit rates may differ.

The latter two are particularly important for teleconferencing applications.

In general, changing to a lower rate can only be accomplished by decoding the bit stream back to images and then re-encoding. It was previously assumed that the delays of the two tandemed pairs of coding and decoding operations must be additive and therefore may become unacceptable in some interactive applications. However, investigations by BT have shown that transcoders can be constructed such that the total delay from the input to the first encoder, through the transcoder, to the output of the final decoder is substantially less than the sum of the delays through two equivalent but independent encode-plus-decode pairs [23]. In fact the level of delay can decrease to that of the original encoder plus decoder.

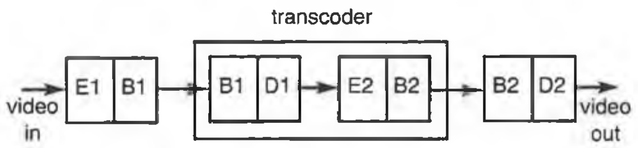


Fig 8 Encoder (E) and decoder (D) kernels and their buffers (B).

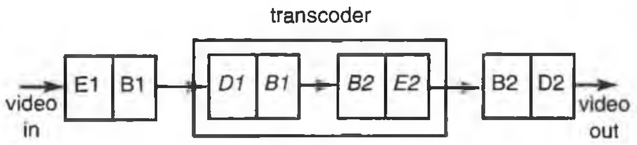
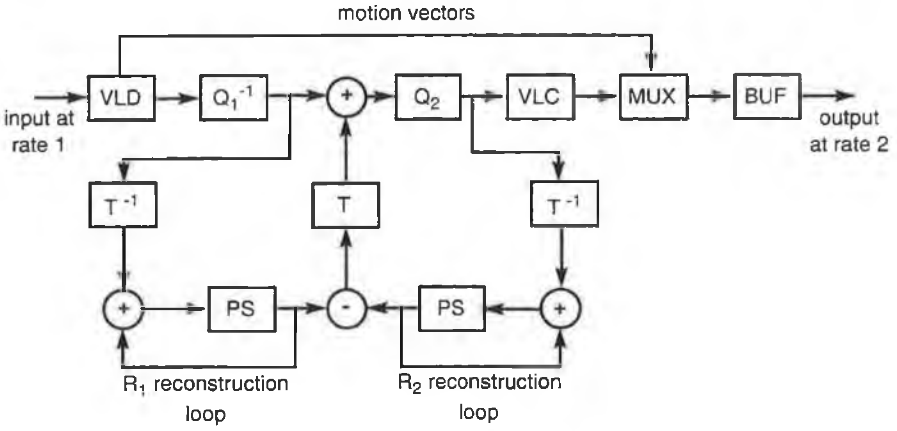


Fig 9 Encoder (E) and decoder (D) with adjacent buffers operating on uncompressed video data.

The key lies in recognising that the vast majority of the unavoidable delay, that which cannot be lessened by more powerful hardware, in a coder plus decoder combination, arises from the rate smoothing buffers. These are needed to match the varying rates of bit generation by the coding kernel and bit consumption by the decoding kernel to constant rate transmission channels. These kernels and buffers are illustrated in Fig 8. In Fig 9 the buffers in the transcoder are rearranged to make them contiguous, by operating on uncompressed instead of compressed video. By precisely controlling the coding kernel in the transcoder so that the fill levels of the two buffers are complementary, the total delay through them becomes constant and they can be removed.

In an extension of this technique the coding kernel in the transcoder is greatly simplified by using the motion vectors from the originating encoder. The bit rate reduction is achieved by requantising of DCT coefficients. However, this causes problems for predictive schemes because the prediction loops at the original encoder and at the final decoder are operating with different data, allowing the loops to diverge and distortion to build up at the final decoder. BT has devised the technique, shown in Fig 10, of introducing a correction signal at the transcoder. This does not eliminate the error but continuously attempts to reduce its size, thus counteracting its growth. The result is a bit stream which has been further compressed, but with very little extra delay, low complexity compared to a full decode and re-encode, and a quality only slightly lower than if the full



VLD	variable length decoder	Q_1^{-1}	inverse quantiser
VLC	variable length coder	Q_2	quantiser
MUX	video multiplexer	BUF	buffer
T	DCT	T^{-1}	IDCT
PS	prediction store (picture store)		

Fig 10 Transcoder for bit rate reduction with drift correction.

compression had been performed in one pass. As the new services mentioned above become more widely used, transcoding will become increasingly commonplace.

5. Future coding algorithms

There is general agreement in the image processing community that existing algorithms are all approaching a limit of compression dictated by purely statistical analysis of images, and exploitation of the human visual system's characteristics. Any large improvements in compression ratios will now require a deeper consideration of the nature of images, in effect using prior knowledge to enable a higher level of abstraction of the image data to be achieved. For example, most images are formed as the projection by a lens of objects in a 3-dimensional world into a 2-dimensional image. There is a direct correspondence between real-world objects and regions in the image. Since the original objects often have some continuity of shade, motion, texture, shape and so on, this continuity is mapped into the resultant image. Conventional coding algorithms such as predictive coding and motion compensation take some advantage of these properties, but do not make full use of the object-like nature of the real world. Admittedly, not everything is 'objects' — water, grass, skin for example are better described as textures, but can usually be segmented into discrete regions in a 2-dimensional image.

Algorithms are being developed which segment each image in a video sequence into regions relating as much as possible to real-world objects, and by analysing the motions of these regions can describe the video sequence in a very compact form [24]. Currently this typically relies on warping the objects in one frame into their new shapes and positions in the next frame, then sending whatever small additional information is necessary to correct any areas not well handled by this method — such as new surfaces revealed by object rotations. It is anticipated that this technique will develop to include full 3-dimensional modelling of the original objects, allowing more accurate reconstruction and higher compression.

Some extreme examples of this form of coding have already been developed in specialised areas. For example, 3-dimensional computer models of human heads have been used to enable synthesis of realistically animated images of human heads talking, moving around naturally, and with facial expressions [25].

Particularly life-like results are achieved if the image of a real person is mapped on to the model. The bit rate required to animate the head is of the order of only a few hundred bits per second. Although synthesis of a human head is now fairly straightforward, analysis of an image of a person's head to extract the motion, mouth shape and facial expression accurately enough to drive the model remains only partly solved.

Current standards work on advanced coding algorithms is centred around MPEG-4.

5.1 MPEG-4

ISO MPEG-4 is an emerging multimedia coding standard that, in addition to higher compression, aims to support new, content-based tools for communication, access and manipulation of digital audiovisual data. As the traditional boundaries between the computer, telecommunications and TV/film industries are blurring, elements that have previously belonged to each of the areas are being introduced into the other two. Also there are major trends towards wireless communications, interactive computer applications and integration of audiovisual data into more and more applications. MPEG-4 is seeking to address the new expectations and requirements emerging from these developments — for example, to exploit the opportunities offered by low-cost, high-performance computing technology and rapid proliferation of multimedia databases.

Early work on requirements for MPEG-4 video identified eight key functionalities which were not thought to be well supported by existing or other emerging standards. These were grouped into the three fundamental categories of 'Content-Based Interactivity', 'Compression Functionality' and 'Universal Access Functionality' (see Table 1).

Table 1 MPEG-4 video categories.

Categories	Functionalities
Content-based interactivity	Content-based multimedia data access tools Content-based manipulation and bit stream editing Hybrid natural and synthetic data coding Improved temporal random access
Compression functionality	Improved coding efficiency Coding of multiple concurrent data streams
Universal access functionality	Robustness in error-prone environments Content-based scalability

The full MPEG-4 standard will comprise six parts, of which systems, audio and visual are the most advanced at the time of writing. Other parts include a conformance testing specification, a technical report on reference software implementations and a multimedia integration framework definition. The systems, audio and visual parts are currently at working draft (WD) status and are expected to progress to full international standard (IS) status in January 1999. Audio (WD 14496-3) and visual (WD 14496-2) will define a standardised coded representation of audio and visual content, both natural and synthetic, called 'audio-visual objects' or AVOs. Systems (WD 14496-1) will standardise the composition of these objects together to form compound AVOs (e.g. an audiovisual scene), and multiplex and synchronise the data associated with individual objects, so that they can be transported over networks at appropriate quality of service levels.

The visual standard will include all the coding tools relating to visual data, natural and synthetic. Currently there are two verification model (VM) specifications — the video VM for natural video coding, frame based or arbitrarily shaped; and the synthetic/natural hybrid coding (SNHC) VM for synthetic 2D/3D graphics tools. As the tools are tested and proven within these VMs, they will be incorporated into the visual WD, and ultimately, if widely agreed on, in the IS.

For the highest efficiency in video compression the video VM is building on the work of ITU-T Recommendation H.263. Extensions have already been made to the core coder, most significantly the ability to code shape and transparency information of so-called video object planes or VOPs (see Fig 11). The video coder will be capable of coding video at rates from 10 kbit/s to more than 4 Mbit/s.

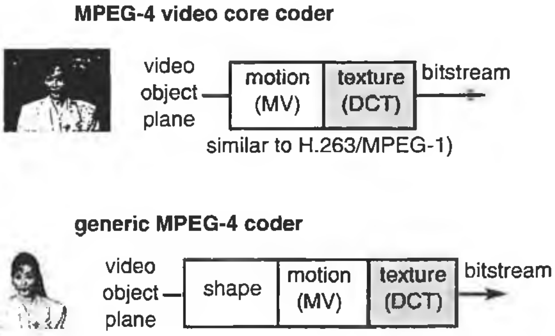


Fig 11 Video verification model stages.

Transcoding between MPEG-4 and MPEG-1/H.263 elementary bit streams should be straightforward in the frame-based (or rectangular VOP) operating mode (e.g. when using the proposed real-time communications profile). The addition of arbitrary shape coding in the generic coder enables a range of new content-based functionalities such as content manipulation in the compressed domain, and content scalability. The shape coding is performed using a bitmap-based technique known as context-based arithmetic coding.

The visual standard will not specify how shape and alpha (transparency information) planes are to be generated as this is a producer or encoder issue. MPEG philosophy is that only decoder issues should be specified to guarantee interworking — enabling competition between companies as to which can provide the best encoder engine, and also, in this instance, provide the most useful shape and alpha data. Blue-screen, or 'chroma-key', techniques, already in widespread use in the television industry, are one source for this information. The video group is considering automatic video segmentation algorithms which may well form an informative annex to the standard to help content producers. Other automatic and semi-automatic segmentation techniques are also being investigated by the European COST211 ter [26].

The synthetic video components of the SNHC VM currently include media integration of text and graphics (MITG), face and body animation, texture coding (generic and view-dependent textures) and static and dynamic mesh coding with texture mapping. MITG provides the capability to overlay and scroll text, images and graphics on coded video backgrounds. Work on this is currently proceeding and is likely to be included soon in the visual WD.

Face animation will allow definition and animation of synthetic 'talking heads' as shown in Fig 12. Only the face definition parameters (FDPs) and the face animation parameters (FAPs) need standardising here. Work at BT Laboratories [25, 27] combining this with texture mapping (Fig 13), has shown how realistic these synthetic personae can actually be. Body animation issues have not been worked on yet but one proposal in this area known as Jack [28] has already been input to the group.

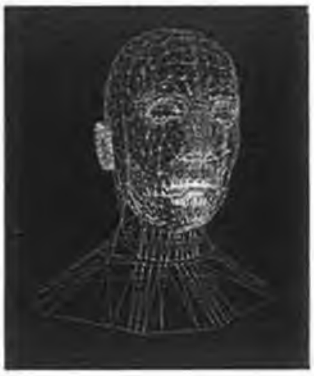


Fig 12 Wireframe head model.

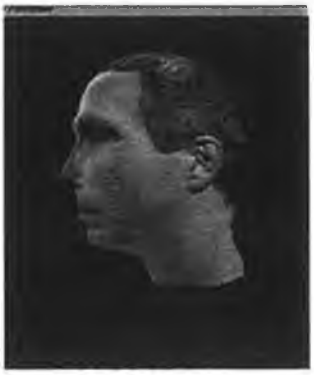


Fig 13 Texture-mapped model.

5.2 MPEG-7

The title of the MPEG-7 activity is 'Multimedia Content Description Interface' [29]. The focus of this new work item is no longer that of efficient *compression* of audio/visual content, but rather its representation for *searching* and *browsing* purposes. This is a very real need with the rapid deployment of digital multimedia databases including sound and images on the World Wide Web (WWW), video-on-demand systems, and corporate image databases — hence the ambitious time-scale of this activity to just one year after ISO/MPEG-4 is finalised.

What MPEG-7 does inherit from previous MPEG activities is the requirement that only the minimum needed for interoperability is to be standardised. For example, it may specify particular sets of audio and video features that can be used as descriptors, such as shape outlines, colour histograms, frequency components, and how they are associated with the original content. However, it will not specify how to extract these video or audio features required for content description, in the same way that the analysis and encoding stages are not specified for the earlier compression-based standards. Nor will it specify a search engine or user interface — which might in practice take the form of a web browser or an intelligent agent. MPEG-7 will concern itself solely with a standardised description of audio/visual content features and will thereby be able to benefit from future progress in automatic feature extraction research. Both information extracted from the scene (e.g. colour, shape, texture) and external information about the scene (e.g. textual annotations, script) are expected to be represented, although the work is at a very early stage at the time of writing.

The standard will have many application domains but a major one is likely to be WWW search engine functionality on audio, video and still pictures. Nearly all WWW searching is currently done on a text basis, although image-based searching is starting to appear in a basic form [30]. MPEG-7 is currently in a requirements capture phase of development but is scheduled to reach 'International Standard' status in the year 2000.

6. Conclusions

Video coding has been one of the success stories of the last decade, enabling a host of applications including PC-based multimedia, videotelephony, highly flexible video-on-demand, and digital broadcasting, that were not possible with the previous analogue formats or uncompressed digital video. The need for compression comes from the limited bit rates of transmission channels, and the cost of digital storage. Although both of these will continue to increase in capacity and reduce in price, we still seem to be some way from the time when the cost will be so low that compression is not needed. Even then, some means of delivery such as radio and satellite channels will continue to have restricted bandwidths.

The successful market development in many areas has been dependent upon agreed standards, principally from the ITU and ISO/MPEG, although particularly for PC-based applications, proprietary algorithms have also found a place.

Conventional compression algorithms seem to be reaching an asymptote, and although MPEG-4 will provide more functionality in terms of being able to define and code objects within a scene separately, in terms of absolute compression it will not provide very much gain over MPEG-2 or H.263. New techniques in MPEG-4 for Synthetic/Natural Hybrid Coding will push compression further, but only in limited domains such as human head/body modelling, and synthetically generated video. Nevertheless, video coding remains an active area of research with, for example, the full potential of wavelet-based schemes yet to be realised, and it is expected that compression ratios will continue to inch upwards.

Using real video in a teleconference has the distinct advantage that within the bounds of coding distortion there is no direct possibility of misrepresentation of the participant's appearance including facial expressions and body language, and brings a feeling of trust to the teleconference. A purely synthetic avatar in a virtual teleconference could be made to do anything, but the complete independence from one's surroundings and the physical world that an avatar brings makes new teleconferencing paradigms possible.

References

1 ITU-R Recommendation BT.601: 'Encoding parameters of digital television for studios'.

2 ITU-T Recommendation V.42 bis: 'Data compression procedures for data circuit terminating equipment (DCE) using error correction procedures'.

3 Ziv J and Lempel A: 'A universal algorithm for sequential data compression', IEEE Transactions on Information Theory, 23, pp 337—343 (May 1977).

4 Jayant N S and Noll P: 'Digital coding of waveforms', Prentice-Hall Signal Processing Series, pp 146—148 (1984).

5 Huffman D: 'A method for the construction of minimum redundancy codes', Proc IRE, pp 1098—1101 (1962).

6 Witten I H, Radford M N and Cleary J G: 'Arithmetic coding for data compression', Communications of the ACM, 30, No 6, pp 520—540 (June 1987).

7 Proceedings of the IEEE, Special issue on Wavelets, 84, No 5 (April 1996).

8 Shapiro J M: 'Embedded image coding using zerotrees of wavelet coefficients', IEEE Trans on Signal Processing, 41, No 12, pp 3445—3462 (December 1993).

9 Gersho A and Gray R M: 'Vector quantisation and signal compression', Kluwer Academic Publications (1992).

10 Barnsley M F and Hurd L P: 'Fractal image compression', A K Peters Ltd, Wellesley, Massachusetts (1991).

11 Beaumont J M: 'Image data compression using fractal techniques', BT Technol J, 9, No 4, pp 93—108 (October 1991).

12 Duffy T S and Nicol R C: 'A codec system for worldwide videoconferencing', Globecom '82, 3, pp 992—997 (1982).

13 ITU-T Recommendation H.261: 'Video codec for autiovisual services at p × 64 kbit/s'.

14 ITU-T Recommendation H.320: 'Narrow-band visual telephone systems and terminal equipment'.

15 International Standard IS 11172-2: 'Coding of moving video and associated audio at rates up to about 1.5 Mbit/s, Part 2: Video'.

16 International Standard IS 13818-2 and ITU-T Recommendation H.262: 'Generic coding of moving pictures and associated audio, Part 2: Video'.

17 Kerr G W: 'A review of fully interactive video on demand', Signal Procecessing: Image Communication, 8, pp 173—190 (1996).

18 ITU-T Recommendation V.34: 'A modem operating at data signalling rates of up to 28 880 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits'.

19 ITU-T Recommendation H.263: 'Video coding for narrow telecommunications channels'.

20 Whybray M W and Ellis W: 'H.263 — video coding recommendation for PSTN videophone and multimedia', IEE Colloquium 95/154 (June 1995).

21 International Standard IS 10918-1 and ITU-T Recommendation T.81: 'Information technology — digital compression and coding of continuous tone still images, Part 1: Requirements and guidelines'.

22 Penbaker W B and Mitchell J L: 'JPEG still image data compression standard', Van Nostrand Reinhold (1992).

23 Morrison G: 'Video transcoders with low delay', IEICE Trans (June 1997).

24 Musmann H G, Hotter M and Ostermann J: 'Object-oriented analysis-synthesis coding of moving images', Signal Processing: Image Communication, 1, No 2, pp 117—138 (October 1989).

25 Welsh W J, Searby S and Waite J B: 'Model-based image coding', BT Technol J, 8, No 3, pp 94—106 (July 1990).

26 Mulroy P: 'Video content extraction: review of current automatic segmentation algorithms', to be presented at COST211 ter Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '97), UCL, Leuven-la-Neuve, Belgium (June 1997).

27 Mortlock A et al: 'Virtual conferencing', BT Technol J, 15, No 4, pp 120—129 (October 1997).

28 'JACK Human Body Modelling Software', Centre for human modelling and simulation, University of Pensylvania, http://www.cis.upenn.edu/~hms/jack.html

29 Koenen R: 'Overview of MPEG-7 goals and objectives', COST 211 ter Workshop in Image Analysis for Multimedia Interactive Services (WIAMIS '97), UCL, Louvain-la-Neuve, Belgium (June 1997).

30 Yahoo Image Surfer, http://isurf.yahoo.com/

Mike Whybray joined BT Laboratories in 1977, working on reliability physics, and in 1983 moved to the visual telecommunications division where he led a group whose activities included developing videophones for deaf people, writing software for DSP-wbased videophones, and building demonstrators of intelligent surveillance systems and synthetic 'talking head' displays. He currently leads a group developing new speech and image coding algorithms and standards. He is a Chartered Engineer and a Member of the IEE.

Geoff Morrison graduated from St John's College, Cambridge in 1971 and joined BT initially working on analogue techniques for video transmission and storage. From 1980 he headed a group which designed, constructed and installed ten analogue and 2 Mbit/s video switches together with the associated software control system for BT's visual services trial. In 1985 he spent 6 months with NTT Labs in Japan. Since then he has been heavily involved directly in the ITU-T and ISO standardisation bodies concerning digital video compression, and in various supporting European collaborative projects.

Patrick Mulroy received a 1st class Honours degree in Electrical and Electomic Engineering from Heriot-Watt University in 1992, and joined the image coding and processing group at BT Laboratories. Here he has worked on image compression algorithms including implementation of H.261 on a parallel TMS320C40 system, development and testing of H.263 (particularly arithmetic coding), and on object segmentation and tracking. He is the BT representative on the COST211 ter project, and MPEG-4 and an associate member of the IEE.