Analysis of Shot Boundary Detection Techniques on a Large Video Test Suite

By

Colin O'Toole

School of Computer Applications

Dublin City University

Glasnevin, Dublin 9,

Ireland.

Supervisor: Prof. Alan Smeaton

A dissertation submitted for the degree of Master of Science

October 1999

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Master of Science in Computer Applications, 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: Colin O'Toole

Date: 31 / 08 / 99

Acknowledgements

I would like to thank my supervisor Dr. Alan Smeaton for his help and guidance during the course of my research. I would also like to thank the other members of the Dublin City University Video Project for their support and assistance, and the National Software Directorate for financial support under contract number SP05-1997.

Thanks to all the members of the six-year club, and my fellow postgrads, for preserving what little sanity I possess. Thanks especially to the members of the MMIR group. Special mention must go to Mighty, NastyBot, Fanis, DoozerBot, and MachoMan without whose intervention this thesis would have long since been finished.

Thanks to Tanya for everything.

Finally, special thanks to my parents and family, whose support and encouragement has made everything possible.

Abstract

This thesis investigates how content-based indexing and retrieval systems can be used to analyse digital video. We focus particularly on the challenge of applying colour-analysis methods to large amounts of heterogeneous television broadcast video.

Content-based systems are those which attempt to automatically analyse image or video documents by identifying and indexing certain features present in the documents. These features may include colour and texture, shape, and spatial locations.

Digital video has become hugely important through the widespread use of the Internet and the increasing number of digital content providers supplying the commercial and domestic markets. The challenge facing the indexing of digital video information in order to support browsing and retrieval by users, is to design systems that can accurately and automatically process large amounts of heterogeneous video. The basic segmentation of video material into shots and scenes is the basic operation in the analysis of video content.

Although many published methods of detecting shot boundaries exist, it is difficult to compare and contrast the available techniques. This is due to several reasons. Firstly, full system implementation details are not always published and this can make recreation of the systems difficult. Secondly, most systems are evaluated on small, homogeneous sequences of video. These results give little indication how such systems would perform on a broader range of video content types, or indeed how differing content types can affect system performance.

As part of an ongoing video indexing and browsing project, our research has focused on the application of different methods of video segmentation to a large and diverse digital video collection. A particular focus is to examine how different segmentation methods perform on different video content types. With this information, it is hoped to develop a system capable of accurately segmenting a wide range of broadcast video.

Other areas addressed in this thesis include an investigation of evaluation methods for digital video indexing systems, and the use of adaptive thresholds for segmentation of video into shots and scenes.

1.0 Introduction

1.1 Introduction

Digital video has become a hugely popular medium for information archiving, storage and delivery. Its potential advantage over traditional analogue formats includes ease of creation, manipulation, and retrieval of content. However this potential is yet to be fully realised due to the limitations of current indexing and retrieval techniques.

The first chapter of the thesis introduces the concepts of digital video, in particular the MPEG-1 format upon which our research was focused. Also presented is a general overview of the indexing and retrieval of digital video.

1.2 Digital Video

Digital video technology has the potential to achieve much higher levels of image quality than its analogue predecessor, and the technology is being improved at an increasing rate. Apart from the improvement in image quality, digital video has a number of unique properties that make possible applications that could not be realised using analogue video. Firstly, digital video can be manipulated more easily than analogue video. In addition to this, digital video can be stored on random access media, whereas analogue video is generally stored sequentially on magnetic tape. This random access allows for increased interactivity, since individual video frames are addressable and can be accessed quickly. Finally, video in digital form can be transmitted across channels unavailable to analogue video. These properties allow the development of applications unique to digital video.

Digital video is more easily manipulated than analogue video. Manipulation of digital video has become so advanced that real life action can be seamlessly merged with computer generated images, as seen in many modern motion frames including Jurassic Park and Independence Day.

Digital video can be duplicated without loss of quality. This property is important for video production and editing applications. Manipulating digital video is not exclusively the preserve of film and television producers however. Desktop video editing is possible on most high end desktop computers and many come with special hardware to digitise video. Using such video applications, users can edit digital video on their desktop to produce multimedia content, which can be integrated into other applications.

The ability to easily store and transmit digital video is perhaps its most important property. It allows users to add video attachments to e-mail, sometimes called v-mail and makes possible video telephony. Video-telephony, or video conferencing, was the first widespread application of digital video. The attraction of video-telephony is due primarily to the high cost of travel, and has spread from the corporate sector to the home market, with the availability of cheap hardware and software solutions.

Because digital video (when compressed) can be transmitted using less bandwidth than analogue television, it is possible to provide many channels where before there were only a few or none. By exploiting this technology, cable TV systems now have enough capacity to provide hundreds of channels of digital video content. These advances have lead to an explosion of digital content providers, originating in America but recently joined by their counterparts in Europe, including the BBC and Sky.

The relatively low bandwidth requirements of digital video also make inexpensive storage a possibility, and digital video can be stored on compact disc. In 1993 Philips introduced movies stored on CD that could be played on Compact Disc Interactive (CD-I) and CD-ROM machines. More recently, the introduction of the Digital Versatile Disk (DVD) has allowed for large amounts of high quality HDTV (High Definition Television) to be stored on a single disk using the MPEG-2 format.

Video-on-demand is a huge potential market that advances in digital video have opened up. It differs from the familiar pay-per-view system (where viewers call up and gain access to a predetermined movie at a set time) in that viewers can choose the programme and the time that they wish to see it. Video-on-demand services may eventually make video rental, as we know it obsolete, but only if the technologies and techniques to allow content access to digital video are improved. These factors are also crucial to Interactive Television, which exploits the random access properties of digital video. Because specific segments of video can be accessed individually, video can appear in response to the viewer's requests. This can be put to good use in educational, training and entertainment applications. The degree of interaction can vary between applications, and can range from choice of program as in video on demand, choice and order of topics as in many training applications, or even continuous interaction as in games.

Multimedia, which concerns the integration of digital video into interactive applications along with other media, such as sound, animation, photographs and text, is one of the most popular applications of digital video. Multimedia applications exploit all of the properties of digital video. The ease with which video can be manipulated allows low cost production of video sequences by non-television professionals. The low bandwidth requirements of digital video allow it to be stored on compact discs or hard disks and to be displayed on computer screens. In addition the ease with which various segments can be accessed allows them to be integrated into highly interactive applications.

1.3 MPEG-1

1.3.1 Introduction

Many of the applications of digital video discussed above rely on compression of the original video. With the increasing commercial interest in video communications, the need for international video compression standards to promote both interoperability and economy of scale is obvious.

The Moving Frame Experts Group (MPEG) was formed in 1988 to develop video coding standards. The first standard developed was ISO/IEC 11172, commonly known as MPEG-1. The main goal was to allow the storage of live video and stereo sound on CD-ROM or CD-I. As the CD products of the time were only single speed (compared to the forty speed models of today), this implied a maximum bit rate of 1.5 Mb/s. This limitation means that MPEG-1 encoded videos sacrifice resolution, generally using the SIF format of 352x288 pixels at 30Hz. As such they are not suited to applications requiring very high resolutions, for example HDTV. The following sections describe the basic operations of the MPEG-1 standard.

1.3.2 General Decoding Process

In its most general form, an MPEG system stream is made up of two layers:

- The system layer contains timing and other information needed to demultiplex the audio and video streams and to synchronise audio and video during playback.

- The compression layer includes the audio and video streams.

Figure 1-1 shows a generalised decoding system for the audio and video streams.

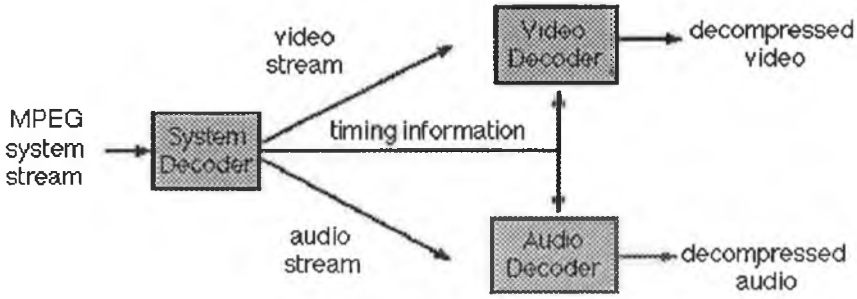


Figure 1-1. General MPEG Decoding System

The system decoder extracts the timing information from the MPEG system stream and sends it to the other system components. (The Synchronisation section has more information about the use of timing information for audio and video synchronisation.) The system decoder also demultiplexes the video and audio streams from the system stream; then sends each to the appropriate decoder.

The video decoder decompresses the video stream as specified in Part 2 of the MPEG standard. (See sections 1.3.4 and 1.3.5 for more information about video compression.) The audio decoder decompresses the audio stream as specified in Part 3 of the MPEG standard. We shall focus on the video aspect of the MPEG-1 standard.

1.3.3 Video Stream Data Hierarchy

The MPEG standard defines a hierarchy of data structures in the video stream as shown schematically in Figure 1-2

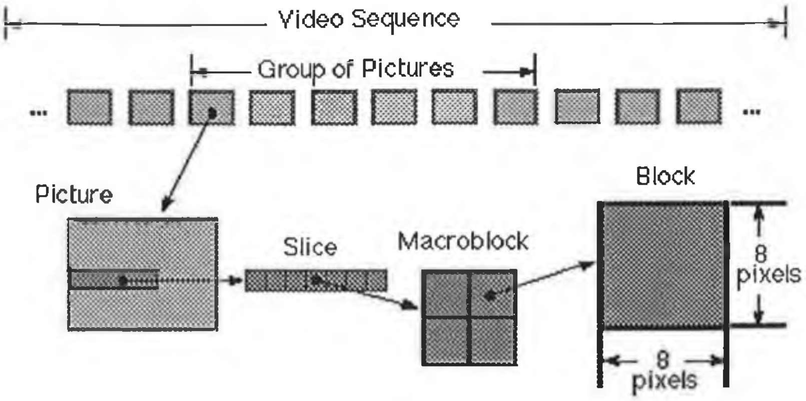


Figure 1-2. MPEG Data Hierarchy

The Elements that compose the MPEG-1 video stream, and are shown in figure 1-2, are listed below:

1.3.3.1 Video Sequence

The video sequence begins with a sequence header, which may contain additional sequence headers. It includes one or more groups of frames, and ends with an end-of-sequence code.

1.3.3.2 Group of Pictures (GOP)

A Group of Pictures consists of a header and a series of one or more frames intended to allow random access into the sequence.

1.3.3.3 Picture (Frame)

The primary coding unit of a video sequence is the picture, or frame. A picture consists of three rectangular matrices representing luminance (Y) and two chrominance (Cb and Cr) values. The Y matrix has an even number of rows and columns. The Cb and Cr matrices are one-half the size of the Y matrix in each direction (horizontal and vertical). The reason for this is that the human eye is more sensitive to luminance than chrominance. By sub-sampling the chrominance values so that the ratio of Y:Cb:Cr is 4:1:1, the MPEG-1 encoder achieves substantial compression without significantly affecting the perceived image quality.

Figure 1-3 shows the relative x-y locations of the luminance and chrominance components. Note that as shown in Figure 1-3, and explained above, every four luminance values have two associated chrominance values: one Cb value and one Cr value. (The location of the Cb and Cr values is the same, so only one circle is shown in the figure.)

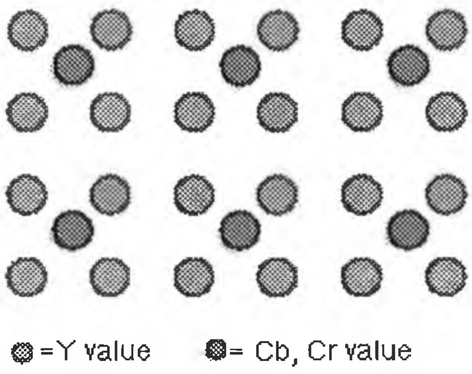


Figure 1-3. Location of Luminance and Chrominance Values

1.3.3.4 Slice

A slice is composed of one or more ``contiguous" macroblocks. The order of the macroblocks within a slice is from left-to-right and top-to-bottom.

Slices are important in the handling of errors. If the bitstream contains an error, the decoder can skip to the start of the next slice. Having more slices in the bitstream allows better error concealment, but uses bits that could otherwise be used to improve picture quality.

1.3.3.5 Macroblock

This is a 16-pixel by 16-line section of luminance components and the corresponding 8-pixel by 8-line section of the two chrominance components. See Figure 1-3 for the spatial location of luminance and chrominance components. A macroblock contains four Y 8x8 pixel blocks, one Cb 8x8 block and one Cr 8x8 block as shown in Figure 1-4. The numbers correspond to the ordering of the blocks in the data stream, with block 1 first.

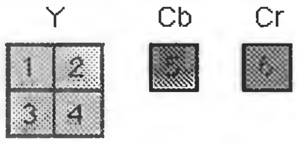


Figure 1-4. Macroblock Composition

1.3.3.6 Block

A block is an 8-pixel by 8-line set of values of a luminance or a chrominance component. Note that due to sub-sampling, a luminance block corresponds to one-fourth as large a portion of the displayed image as does a chrominance block. It is important to realise that MPEG is a block-based encoding/decoding scheme, and not a frame based one. This will become apparent as we consider the methods of inter-frame coding in section 1.3.5.

1.3.4 Intra frame coding

This section describes the methods by which MPEG eliminates spatial redundency in single frames. These techniques affect only one frame, therefore they are referred to *as intra frame coding* (or transform coding) techniques. MPEG also employs *Inter frame coding* techniques to eliminate redundency between frames - these techniques are discussed in section 1.3.5. The MPEG transform coding algorithm includes these steps:

- Discrete cosine transform (DCT)
- Quantization
- Run-length encoding

Both image blocks and prediction-error blocks have high spatial redundancy. To reduce this redundancy, the MPEG algorithm transforms 8 x 8 blocks of pixels or 8 x 8 blocks of error terms from the spatial domain to the frequency domain with the Discrete Cosine Transform (DCT).

Next, the algorithm quantizes the frequency coefficients. Quantization is the process of approximating each frequency coefficient as one of a limited number of allowed values. The encoder chooses a quantization matrix that determines how each frequency coefficient in the 8 x 8 block is quantized. Human perception of quantization error is lower for high spatial frequencies, so high frequencies are typically quantized more coarsely (i.e., with fewer allowed values) than low frequencies. Quantization results in a reduction of the original image quality, as when the image is reconstructed upon decoding, the pixel values will have been altered by the approximation process. Due to this and other techniques employed, MPEG-1 is considered a "lossy" algorithm, i.e. some of the original image quality is lost in the encoding/decoding process.

The combination of DCT and quantization results in many of the frequency coefficients being zero, especially the coefficients for high spatial frequencies. To take maximum advantage of this, the coefficients are organised in a zigzag order to produce long runs of zeros (see Figure 1-5). The coefficients are then converted to a series of run-amplitude pairs, each pair indicating a number of zero coefficients and the amplitude of a non-zero coefficient. These run-amplitude pairs are then entropy coded using a variable-length code (VLC) table, which uses shorter codes for commonly occurring pairs and longer codes for less common pairs.

Some blocks of pixels need to be coded more accurately than others. For example, blocks with smooth intensity gradients need accurate coding to avoid visible block boundaries. To deal with this inequality between blocks, the MPEG algorithm allows the amount of quantization to be modified for each macroblock of pixels. This mechanism can also be used to provide smooth adaptation to a particular bit rate.

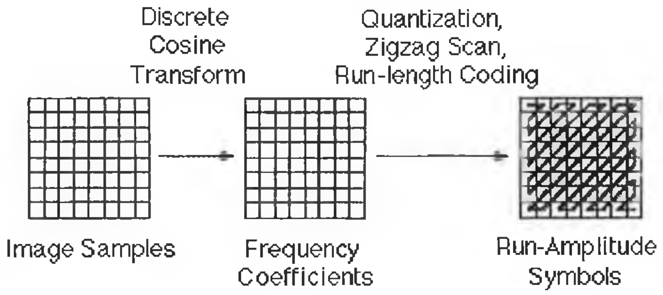


Figure 1-5. Transform Coding Operations

1.3.5 Inter frame coding

Much of the information in a frame within a video sequence is similar to information in a previous or subsequent frame. The MPEG standard takes advantage of this *temporal redundancy* by representing some frames in terms of their differences from other (reference) frames, or what is known as *inter-frame* coding. This section describes the types of coded frames and explains the techniques used in this process.

The MPEG standard specifically defines three types of frames: intra (I), predicted (P), and bidirectional (B).

1.3.5.1 Intra Frames

Intra frames, or I-frames, are coded using only information present in the frame itself. I-frames provide potential random access points into the compressed video data. I-frames use only intra-frame (transform) coding (as explained in section 1.3.4) and as such only provide moderate compression.

1.3.5.2 Predicted Frames

Predicted frames, or P-frames, are coded with respect to the nearest previous I- or P-frame. This technique is called forward prediction and is illustrated in Figure 1-6.

Like I-frames, P-frames serve as a prediction reference for B-frames and future P-frames. However, P-frames use motion compensation (see Motion Compensation below) to provide more compression than is possible with I-frames. Unlike I-frames, P-frames can propagate coding errors because P-frames are predicted from previous reference (I- or P-) frames.

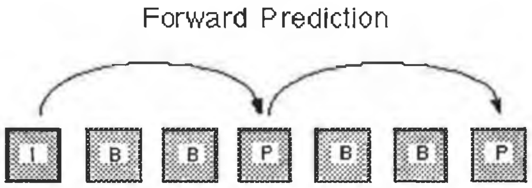


Figure 1-6. Forward Prediction

1.3.5.3 Bidirectional Frames

Bidirectional frames, or B-frames, are frames that use both a past and future frame as a reference. This technique is called bidirectional prediction and is illustrated in Figure 1-7. B-frames provide the most compression and do not propagate errors because they are never used as a reference. Bidirectional prediction also decreases the effect of noise by averaging two frames.

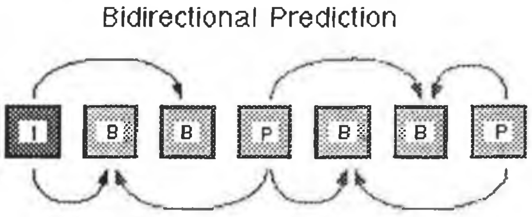


Figure 1-7. Bidirectional Prediction

Frequency and location of frame types

The MPEG algorithm allows the encoder to choose the frequency and location of I-frames. This choice is based on the application's need for random accessibility and the location of scene cuts in the video sequence. In applications where random access is important, there would be two I-frames per second of video (25 or 30 frames depending on whether the broadcast format is PAL or NTSC).

The encoder also chooses the number of B-frames between any pair of reference (I- or P-) frames. This choice is based on factors such as the amount of memory in the encoder and the characteristics of the material being coded. For example, a large class of scenes have two bidirectional frames separating successive reference frames. A typical arrangement of I-, P-, and B-frames is shown in Figure 2-8 in the order in which they are displayed.

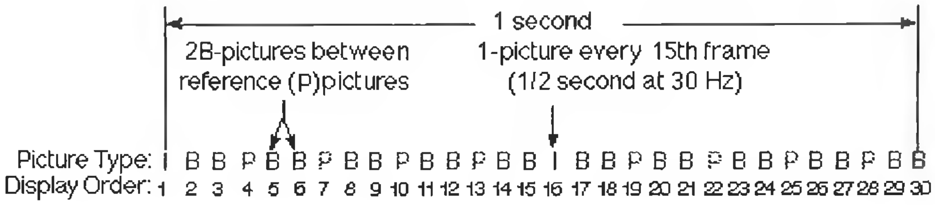

1 second
2B-pictures between reference (P)pictures
1-picture every 15th frame (1/2 second at 30 Hz)
Picture Type: I B B P B B P B B P B B P B B I B B P B B P B B P B B P B B
Display Order: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Figure 1-8. Typical Display Order of Frame Types

The MPEG encoder reorders frames in the video stream to present the frames to the decoder in the most efficient sequence. In particular, the reference frames needed to reconstruct B-frames are sent *before* the associated B-frames. Figure 1-9 demonstrates this ordering for the first section of the example shown above.

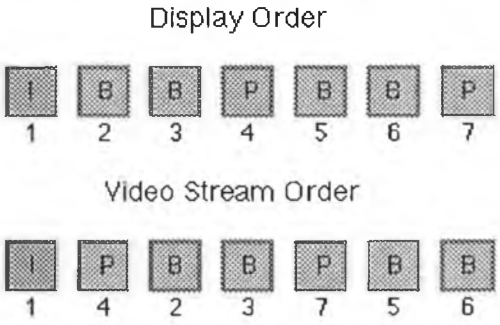


Figure 1-9. Video Stream versus Display Ordering

Motion Compensation

Motion compensation is a technique for enhancing the compression of P- and B-frames by eliminating temporal redundancy. Motion compensation typically improves compression by about a factor of three compared to intra-frame coding. Motion compensation algorithms work at the macroblock level. When a macroblock is compressed by motion compensation, the compressed file contains this information:

- The spatial vector between the reference macroblock(s) and the macroblock being coded (motion vectors)
- The content differences between the reference macroblock(s) and the macroblock being coded (error terms)

Not all information in a frame can be predicted from a previous frame. Consider a scene in which a door opens: The visual details of the room behind the door cannot be predicted from a previous frame in which the door was closed. When a case such as this arises - i.e., a macroblock in a P-frame cannot be efficiently represented by motion compensation - it is coded in the same way as a macroblock in an I-frame using transform coding techniques (see section 1.3.4).

The difference between B- and P-frame motion compensation is that macroblocks in a P-frame use the previous reference (I- or P-frame) only, while macroblocks in a B-frame are coded using any combination of a previous or future reference frame.

A result of this is that B frames may contain a mix of I, B and P macroblocks, depending on the amount of visual details that can be predicted from reference frames. Four codings are therefore possible for each macroblock in a B-frame:

- Intra coding: no motion compensation
- Forward prediction: the previous reference frame is used as a reference
- Backward prediction: the next frame is used as a reference
- Bidirectional prediction: two reference frames are used, the previous reference frame and the next reference frame

Backward prediction can be used to predict uncovered areas that do not appear in previous frames.

1.4 Video indexing and retrieval

The indexing and retrieval of digital video is an active research area in computer science. The increasing availability and use of on-line video has led to a demand for efficient and accurate automated video analysis techniques. As a basic, atomic operation on digital video, much research has focused on segmenting video by detecting the boundaries between camera shots.

A *shot* may be defined as a sequence of frames captured by "a single camera in a single continuous action in time and space" [3]. For example, a video sequence showing two people having a conversation may be composed of several close-up shots of their faces which are interleaved and make up a scene. Shots define the low-level, syntactic building blocks of a video sequence.

A large number of different types of boundaries can exist between shots [8]. A *cut* is an abrupt transition between two shots that occurs between two adjacent frames. A *fade* is a gradual change in brightness, either starting or ending with a black frame. A *dissolve* is similar to a fade except that it occurs between two shots. The images of the first shot get dimmer and those of the second shot get brighter until the second replaces the first. Other types of shot transitions include wipes and computer generated effects such as morphing.

A *scene* is a logical grouping of shots into a semantic unit. A single scene focuses on a certain object or objects of interest, but the shots constituting a scene can be from different angles. In the example above the sequence of shots showing the conversation would comprise one logical scene with the focus being the two people and their conversation.

Figure 1-10 shows the basic composition of an arbitrary video sequence. The sequence is composed of one or more high-level semantic scenes. It is at this level that humans visualise video, rather than at the lower levels which video segmentation algorithms currently operate upon. Each scene is composed of one or more shots, separated by shot boundaries. Our research has focused on automatically detecting these boundaries. Finally, the lowest level primitives in the video sequence are the individual frames.

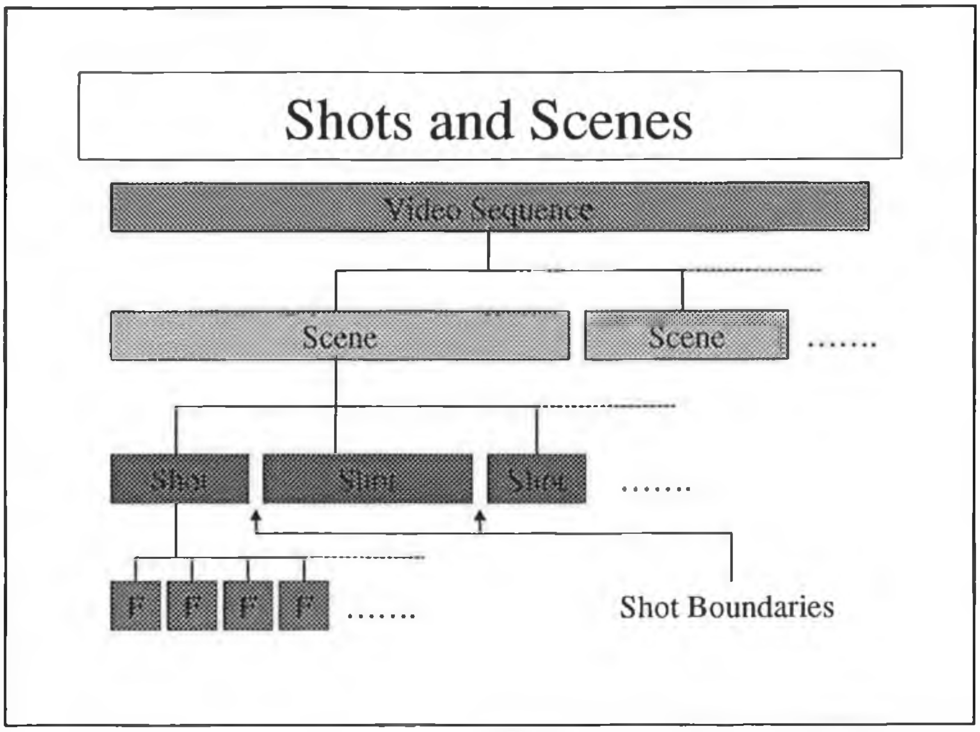


Figure 1-10. Composition of video sequence

The segmentation of video into scenes is far more desirable than simple shot boundary detection. This is because, as mentioned above, people generally visualise video as a sequence of scenes not of shots, just like a play on a stage, and so shots are really a phenomenon peculiar to only video. Scene boundary detection requires a high level semantic understanding of the video sequence and such an understanding must take cues from, amongst other things, the associated audio track and the encoded data stream itself. Shot boundary detection, however, still plays a vital role in any video segmentation system, as it provides the basic syntactic units for higher level processes to build upon.

Our research has focused on accurate segmentation of digital video into shots. To achieve this we detect shot boundaries by measuring colour similarities between video frames. Chapters 3 to 6 contain detailed descriptions of the systems developed.

1.5 Motivation and objectives

Although many published methods of detecting shot boundaries exist, it is difficult to compare and contrast the available techniques. This is due to several reasons. Firstly, full system implementation details are not always published and this can make recreation of the systems difficult. Secondly, most systems are evaluated on small, homogeneous sequences of video. These results give little indication how such systems would perform on a broader range of video content types, or indeed how differing content types can affect system performance.

As part of an ongoing video indexing and browsing project, our research has focused on the application of methods of video segmentation to a large and diverse digital video collection. The aim is to examine how segmentation methods perform on different video content types. With this information, it is hoped to develop a system capable of accurately segmenting a wide range of broadcast video.

This thesis focuses on the results obtained using a video indexing system based on colour histogram comparison and colour moments. Multiple versions of the systems are presented, ranging from basic models similar to those described in [3], to extensions of this basic model, which attempt to adapt to the varied content types found in broadcast television video.

1.6 Summary

In this first chapter we introduced the concepts of digital video and the MPEG-1 encoding standard. The need for compression standards for digital video was explained and an overview of the methods by which MPEG-1 operates was presented. The basic concepts and terms associated with video indexing and retrieval were introduced along with an explanation of the motivation and purpose of the research undertaken. In particular we highlighted the importance of adequate testing of video shot boundary detection methods, using, if possible, a large and heterogeneous test suite, to better approximate the variety of content types found in real world broadcast television.

The next chapter discusses related work in the field of shot boundary detection for digital video. Various methods of shot boundary detection are described in detail, as well as related work in the field of video indexing. Also presented are examples of real world systems that utilise these techniques.

2.0 Related work

2.1 Introduction

The indexing and retrieval of digital video is an active research area in computer science. Many techniques for shot detection have been presented. Some of these methods rely on the method in which the video is compressed, while others are format-independent and are suitable for use on any type of digital video.

There also exists a large number of video systems, both operational and in development, that are based on one or more of these techniques. This chapter firstly presents the various analysis techniques for detecting shot boundaries and then describes a selection of video systems.

2.2 Methods of shot boundary detection

2.2.1 Pixel comparison

Simple pixel comparison was one of the first methods used for detecting frame similarities. In its most basic form, a count of the number of pixels that change between two frames is compared against some pre-determined threshold. If the pixel count exceeds the threshold then a shot boundary is assumed. A number of variations on this basic method exist, mainly with the aim of reducing noise caused by camera motion and other video effects [11, 16].

Mathematically, equations (1) and (2) are used to represent the pixel difference and threshold calculations [1]. In equation (1), $F_i(x, y)$ is the intensity value of the pixel at co-ordinates (x, y) in frame i. If the difference between corresponding pixels in two consecutive frames is above a specified intensity threshold, then the Difference Picture, $DP_i(x, y)$, is set to one. In Equation (2), the difference pictures are summed and divided by the total number of pixels in a frame. If this percentage difference exceeds a certain threshold T, a shot boundary is declared.

$$DPi(x,y) = \begin{cases} 1 \quad if \quad | Fi(x,y) - Fi-1(x,y) | > T \\ 0\, otherwise \end{cases} \qquad (1)$$

$$\frac{\sum_{x,y=1}^{X,Y} DPi(x,y)}{X * Y} * 100 > T \qquad (2)$$

Zhang et al [31] use a pixel-based difference method and also compare mean intensity values for connected pixels, which, although slow, produced good results once the threshold was manually tailored to the video sequence. However, this technique shares the weaknesses of all pixel-based techniques, being sensitive to fast moving objects and global camera motion.

2.2.2 Statistical comparison

Detecting transitions at the pixel level is not very robust, as camera effects results in a large number of false positives. Statistical comparison techniques [11, 31] divide frames into blocks, which are then compared on the basis of statistical comparisons of their intensity levels [1]. Typically, the mean and the variance of blocks in consecutive frames are compared, and a threshold is declared if a certain number of blocks exceed a pre-set threshold value. This approach is generally superior to pixel-based comparisons, although more complicated versions can be computationally expensive.

A likelihood ratio approach has been suggested based on second-order statistics [31]. Equation 3 calculates the likelihood function, which is used to compare video frames based on their pixel intensity levels. Let μ_i and μ_{i+1} be the mean intensity values for a given region in two consecutive frames, and σ_i and σ_{i+1} be the corresponding variances. The number of blocks that exceed a certain intensity threshold value t, are counted. If this number exceeds a certain value then a shot boundary is declared.

$$\lambda = \frac{\left[((\sigma i + \sigma i + 1)/2)^2 + ((\mu i - \mu i + 1)/2)^2\right]^2}{\sigma i * \sigma i + 1} \qquad (3)$$

$$DPi(k,l) = \begin{Bmatrix} 1 \quad if \quad \lambda > t \\ 0\, otherwise \end{Bmatrix}$$

This method is more resistant to object and camera movement than pixel-based comparisons. However, it is possible that two perceptually different blocks will have the same likelihood ratio, and so no change will be detected.

Colour moments have also been used to represent a colour distribution. This is possible because a probability distribution is uniquely characterised by its moments [1, 23]. An implementation of this technique has been incorporated into our research and is discussed fully in chapter 4.

2.2.3 Histogram comparison

Histograms are perhaps the most widely used method of detecting shot boundaries. The basic histogram method creates greyscale or colour histogram signatures for each frame, then computes the bin-wise difference. If this difference is above a pre-defined threshold then a shot boundary is assumed. Histograms are generally more effective than simple pixel comparisons, and considerably faster than equivalent statistical methods.

Nagasaka and Tanaka [20] employed the χ^2 test to compare differences for both grey level and colour histograms. The function, shown in equation 4, uses the square of the difference between the two histograms to strongly reflect any dissimilarity. However, this emphasis on difference also enhances small changes due to object and camera motion.

$$\sum \frac{|Hi(j) - Hi+1(j)|^2}{Hi+1(j)} \tag{4}$$

Gradual shot boundaries (e.g. fades, dissolves and wipes) are difficult to detect using intensity and colour based methods. This is because the gradual change between frames often remains too low to be detected by the thresholds employed by these methods. Zhang et al [31] used a running histograms method in an attempt to detect gradual as well as abrupt shot boundaries. This method calculates the cumulative difference between frames for gradual transitions. Two threshold values are required. The upper threshold is used to detect abrupt shot boundaries, and the lower one to detect gradual transitions. Any frame that exceeds the lower difference threshold is treated as a potential start of a gradual transition. The difference value of each subsequent frame that exceeds the lower threshold is summed. This process continues until either the combined values exceed the upper threshold, and a gradual transition is declared, or a certain number of frames fall below the lower threshold, in which case the potential starting frame is discarded and the process starts again. This technique improved the detection of gradual transitions, but at the cost of increasing the number of false shot boundaries detected by the system (recall is generally improved at the expense of precision). Figure 2-1 graphically describes the operation of the running histograms method.

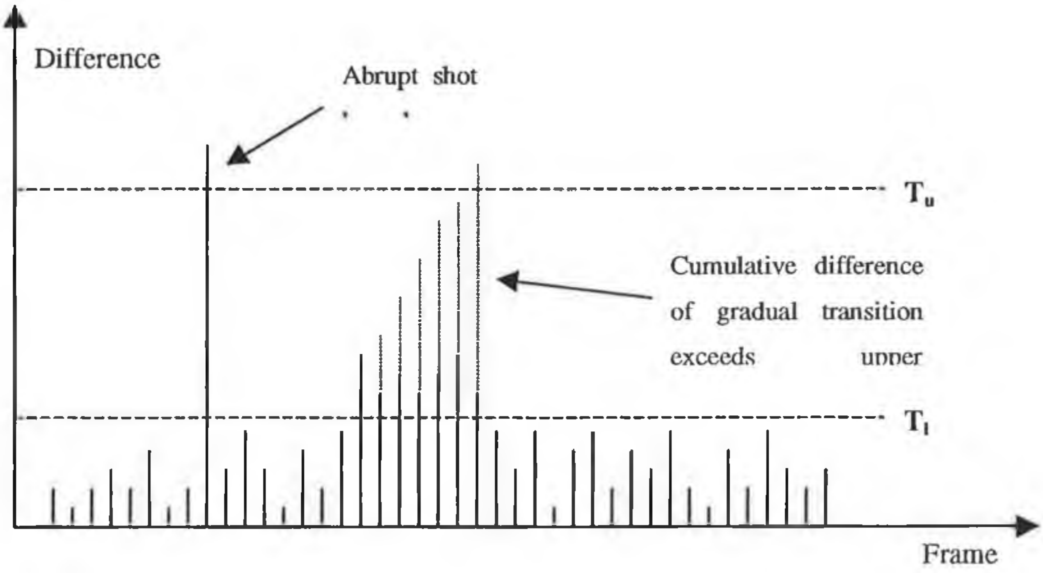


Figure 2-1. Running histograms

Cabedo and Bhattacharjee [3] used the cosine measure for detecting histogram changes in successive frames and found it more accurate than other, similar methods, including the χ^2 test. This method considers the histograms of consecutive frames as N-dimensional vectors, where N is the number of bins in each histogram. Given that a_i is one bin in histogram A and b_i is the corresponding bin in histogram B, the distance measure D_{cos} between these two histograms is then defined as:

$$D_{\cos}(A,B)=1-\frac{\sum_{i=i}^{N}(a_i \cdot b_i)}{\sqrt{\sum_{i=1}^{N} a_i^2 \cdot \sum_{i=1}^{N} b_i^2}}$$

Gong et al [12] used a combination of global and local histograms to represent the spatial locations of colour regions.

2.2.4 Edge detection and tracking

This method involves using a filtering method such as Sobel filtering to produce an edge image of each frame pair. The edge images are then compared and evaluated. Canny [4] suggested the replacement of Sobel filtering with more robust methods, with the aim of defining edges more clearly, particularly in very bright or dark scenes. Zabih et al [30] compared the number and position of edges in successive video frames, allowing for global camera motion by aligning edges between frames. Transitions can be detected and classified by examining the percentages of entering and exiting pixels. The system also employs motion compensation to overcome the effects of global camera motion.

2.2.5 Compression Features

These techniques rely on features of the encoding standard to detect shot boundaries. The advantage of such techniques is that they do not require that the video stream be fully decoded, so the amount of data to be processed is reduced considerably. As MPEG-1 is presently the most popular standard for digital video [25], the majority of these methods utilise its features, as described in chapter 1. Meng et al [19] employ a variety of methods utilising motion vector and DCT co-efficient analysis, including examining the ratio of intracoded and predicted macroblocks in MPEG P-frames to decide if a transition has taken place. Patel and Sethi [21] attempted to detect shot boundaries by examining the average colour histograms of successive I-frames. Their technique delivered reasonable precision but poor recall. Cabedo and Bhattacharjee [3] extended the work of [21] by using a variety of methods to process I, B, and P frames in an MPEG-2 video stream. They found that such techniques delivered good results over a limited test set.

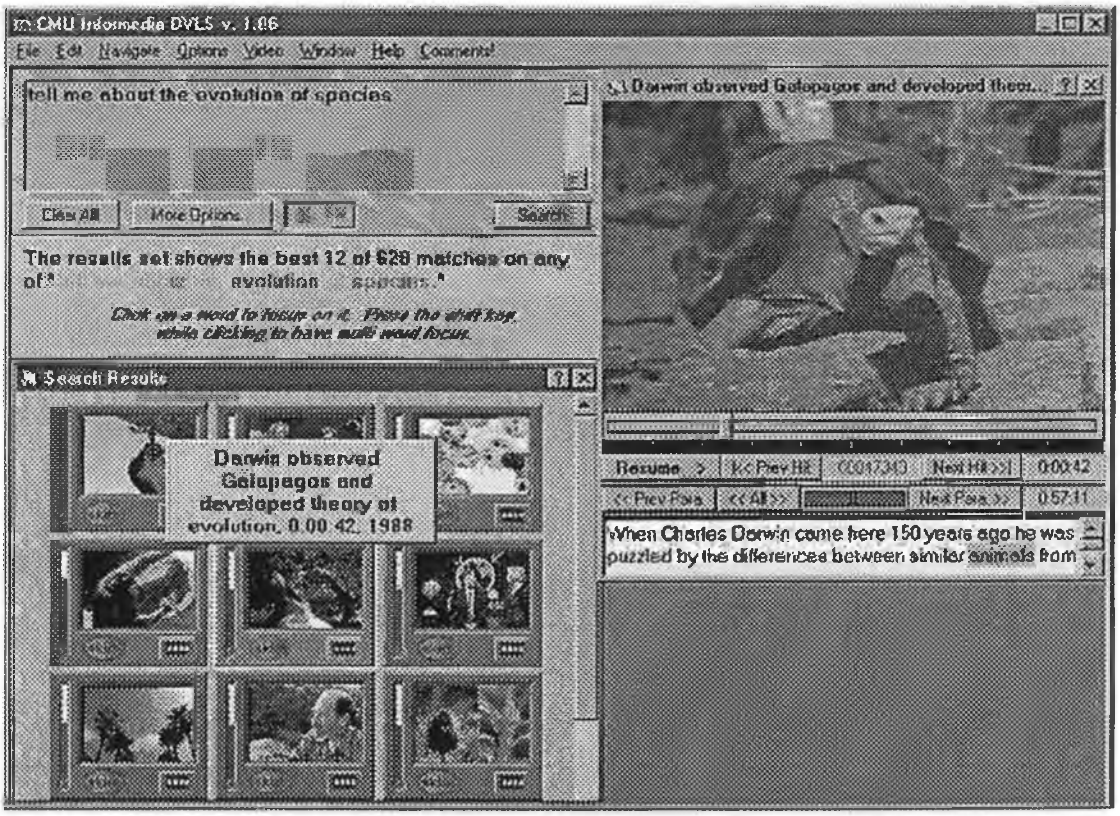
2.3 Existing video indexing and retrieval systems

2.3.1 Informedia (Carnegie Mellon University)/MediaKey (ISLIP Media)

Informedia [8, 9, 10, 11, 28] is possibly the largest video project currently in existence. It's stated aim is to develop new technologies for data storage, search, and retrieval, and embed them in a video library system for use in education, training, sports and entertainment. It is anticipated that the primary media-server file system will require one terabyte (1,000 gigabytes) of storage to archive the 1000 hours of video, taken from WQED Pittsburgh, Fairfax Co. VA School's Electronic Field Trips, and the British Open University's BBC-produced video courses.

The system uses the a speech recogniser to automatically create a transcript from the video soundtrack track, this transcript is then searched using a text information retrieval system. Also supported is the creation of video abstracts to support accelerated video browsing. Figure 2-1 shows a sample screen from the Informedia interface.

MediaKey is the commercial version of the Informedia project. The product employs the same technologies as Informedia, including speech recognition, automatic transcript generation using natural language processing techniques, and a variety of accelerated video browsing techniques. However, like Informedia, initial queries must be textual only, and no content-based query by motion or visual properties is allowed. The interface is similar to that shown in figure 2-2.


CMU Informedia DVLS v. 1.06
File Edit Navigate Options Video Window Help Comments!
Clear All
More Options
Search
Search Results
Darwin observed Galapagos and developed theor...
Resume >
|<Prev Hit
Next Hit>>|
0:00:42
<< Prev Para
<< All >>
Next Para >>
0:57:11

tell me about the evolution of species

The results set shows the best 12 of 628 matches on any of "... evolution ... species."

Click on a word to focus on it. Press the shift key while clicking to have multi word focus.

Darwin observed Galapagos and developed theory of evolution, 0:00:42, 1988

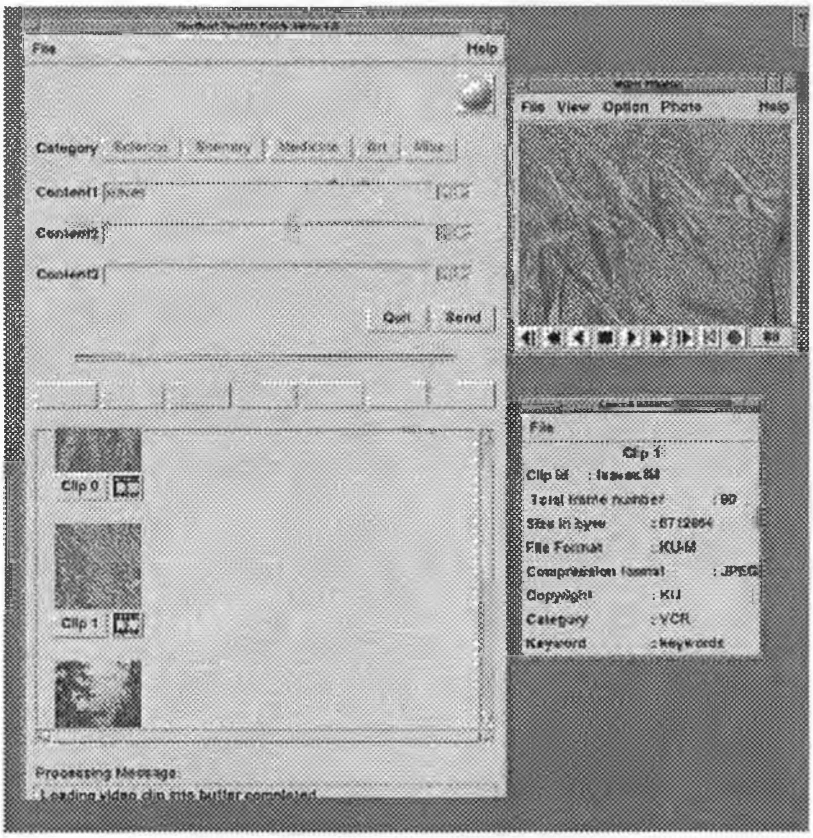
When Charles Darwin came here 150 years ago he was puzzled by the differences between similar animals from

Figure 2-2. Interface for the Informedia system

2.3.2 VISION (University of Kansas)

VISION (Video Indexing for SearchIng Over Networks) [17] is a prototype system for indexing and retrieval of JPEG video and audio using the internet. The system digitises videos, soundtracks and closed-captions in real time, automatically adjusting quality to match any network bandwidth limitations present. The idea behind the approach used is that current systems divide video into too many shots, irrespective of whether the content of the shots is related. An example is a news anchorperson speaking over a number of shots depicting some news event. As the shots all relate to the same event (pointed to by their similar audio characteristics), they should be merged into one scene. So the system segments video in two steps:

1. Segmentation of video into shots
2. Merging of shots into "scenes" or semantic units by their audio characteristics.



Clip 1
Clip Id : [illegible]
Total frame number : 90
Size in byte : 8712864
File Format : KU-M
Compression format : JPEG
Copyright : KU
Category : VCR
Keyword : keywords

Processing Message:
Loading video clip into buffer completed.

Figure 2-3. Video search in the VISION system

The index information is stored in a text-based information retrieval system. Oddly enough given the complexity of the indexing process, the database will only support Boolean text queries, using pre-defined keywords. No facility for free-text queries, sketching, or query by example is provided. The system will be tested on a database of five hours of educational videos. Figure 2-3 shows a sample screen from the VISION interface.

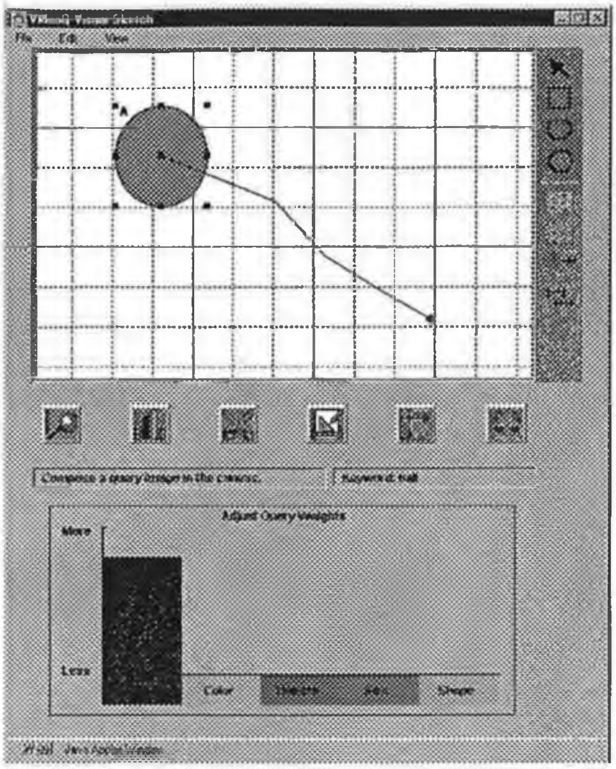
2.3.3 VideoQ (Colombia University)

The VideoQ system [6] is interesting because it expands the traditional textual query/static query sketch to allow spatio-temporal video object queries. This allows the user to sketch objects (shape, texture and colour) and then define their motion trajectories. The system also supports textual queries (through manual annotation) and browsing of the video database. The system focuses on the idea of using animated sketches to formulate queries.

Although this is a powerful paradigm for certain types of queries, the authors freely acknowledge that the system is unsuitable for video with complicated, moving backgrounds. The database is arranged somewhat differently to other systems in that video is stored and retrieved as shots, rather than larger video segments. No information is given on the method of extracting shots from the original video material, this may be a manual process.

Currently the system indexes over 2000 shots. Each shot is compressed and stored in three layers to meet different bandwidth requirements. In addition to query by sketch, the user can browse the video shots or search video by text. The video shots are catalogued into a subject taxonomy, which the user can easily navigate. Each video shot has also been manually annotated so the user can perform simple text search of keywords.

Figure 2-4 shows a sample screen from the VideoQ interface.



Compose a query image in the canvas.

Keyword: ball

Figure 2-4. Defining motion trajectories in the VideoQ system

2.3.4 WebSEEk (Columbia University)

Also from the Columbia stable is WebSEEk [7, 27], an integrated image and video content-based search and browse system. The system locates image and video files on the internet using web robots, and then indexes them. Textual annotation is performed automatically by comparing the directory/file name of the image/video to a concept hierarchy. This hierarchy is also used to ease searching by breaking the database into smaller domains (similar to the Yahoo web search engine). Initial searching is via keywords. The results of this primary search then acts as the basis for subsequent searches using a "find me more like this" search/browse session, which is content based.

One advantage of WebSEEk is its comprehensive query reformulation. This feature allows modification of the query image's histogram, positive and negative feedback on retrieved video clips, as well as more standard reformulation of text queries. An interesting feature is the use of moving thumbnails, where quickly alternating key frames are used to represent a video clip. Figure 2-5 shows a sample screen from the WebSEEk interface.

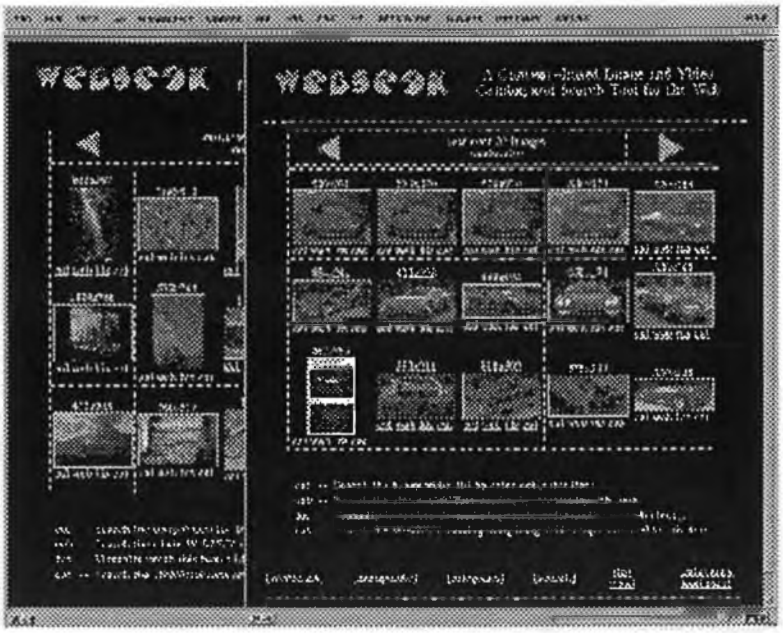


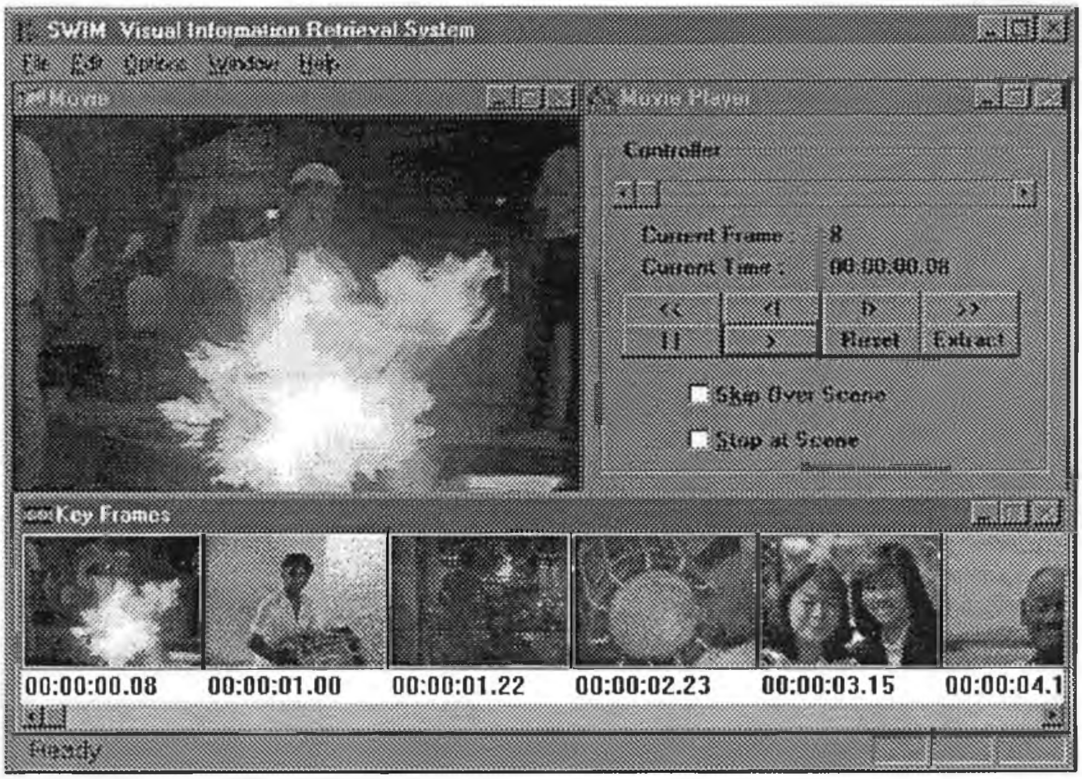
Figure 2-5. Query results in the WebSEEk system

2.3.5 SWIM (Kent Ridge Digital Labs)

The Show What I Mean (SWIM) system [32] has been developed using the KRDL video indexing and retrieval toolkit. A feature of the system is the extensive query options, which include the following:

- Search by browsing pre-defined subject categories.
- Standard keyword query, including an option for including automatic synonym expansion.
- Key frame-based query using content-based techniques including colour, texture, and shape.
- A unique "shot-based" query, which operates on a video shot's inherent temporal properties. These include camera motion and variation of colour and brightness with respect to time.

This system provides automatic shot detection and shot classification according to global camera movement, as well as retrieval by visual features. Browsing of automatically extracted key frames is by a hierarchical interface, or standard key frame list. Figure 2-6 shows an example of the standard key frame interface used to navigate through a video segment.



Current Frame : 8

Current Time : 00:00:00.08

Figure 2-6. Key frame navigation in the SWIM system

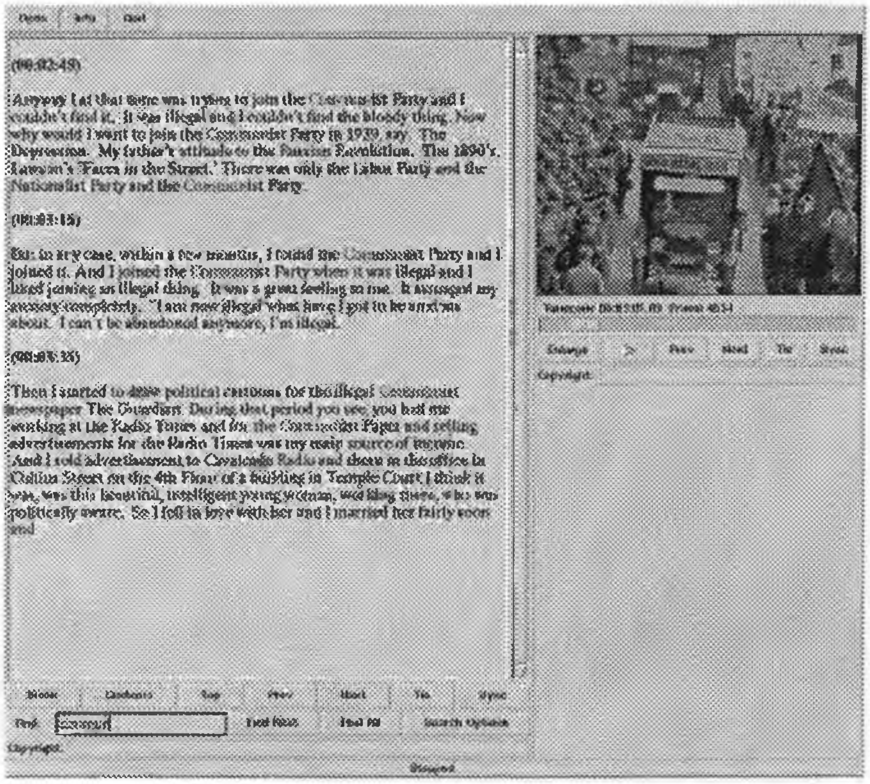
2.3.6 FRANK (CSIRO)

The FRANK (Film/TV Researchers Archival Navigation Kit) [26] is a trial system developed to facilitate the remote access and navigation of video archives. The enabling technology for the trial is an experimental city-to-city ATM network, which allows large amounts of high quality video to be transmitted. The emphasis of the project is on alternative representations of video, such as transcripts and shot-lists. Also, the navigation (search and browse) system and the video server itself are totally separate, thus facilitating a division of labour between the video provider, and the access services for that provider.

The system does not provide an automatic transcript generation, as these have been provided as part of the video database. Search is by keywords only, no visual search option is provided. Once a video clip has been retrieved it may be abstractly viewed in one of two ways:

1. Transcript with synchronised playback of associated video
2. Transcript and synchronised storyboard (list of key frames from each shot)

The project emphasis on network technology means that each of these video abstraction methods may reside different locations and be serviced by different organisations. The apparent strength of this system lies not it's basic querying engine, but rather it's effective and configurable in-clip browsing facilities. Figure 2-7 shows an example of in-clip browsing using the transcript and video playback.



(00:02:45)

Anyway I at that time was trying to join the Communist Party and I couldn't find it. It was illegal and I couldn't find the bloody thing. Now why would I want to join the Communist Party in 1939, say. The Depression. My father's attitude to the Russian Revolution. The 1890's. Lawson's 'Faces in the Street.' There was only the Labor Party and the Nationalist Party and the Communist Party.

(00:03:15)

But in any case, within a few months, I found the Communist Party and I joined it. And I joined the Communist Party when it was illegal and I liked joining an illegal thing. It was a great feeling to me. It assuaged my anxiety completely. "I am now illegal what have I got to be anxious about. I can't be abandoned anymore, I'm illegal.

(00:03:35)

Then I started to draw political cartoons for the illegal Communist newspaper The Guardian. During that period you see, you had me working at the Radio Times and for the Communist Paper and selling advertisements for the Radio Times was my main source of income. And I sold advertisement to Cavalcade Radio and there in the office in Collins Street on the 4th Floor of a building in Temple Court I think it was, was this beautiful, intelligent young woman, working there, who was politically aware. So I fell in love with her and I married her fairly soon and

Figure 2-7. In-clip navigation in the FRANK system

2.4 Summary

This chapter detailed the various methods employed to index and retrieve digital video. As well as a description of general methods employed, details of concrete systems were also included. An examination of these systems is useful to see how abstract techniques translate to real world systems.

Each method described may well perform well in certain situations. There is no generally accepted "best" method that gives superior results for all video types. Typically, each method exhibits strengths and weaknesses, and is suitable for a specific type of video or application.

The next chapter focuses on the video test suite developed and employed during our research. It details the both the rationale behind the development of the test suite, and the details of its implementation.

3.0 The Video Test Suite

3.1 Introduction

Although some published methods of detecting shot boundaries in digital video exist, it is often difficult to compare and contrast available techniques. This is due to the fact that the majority of systems are evaluated using a test suite consisting of small amounts of homogeneous video. Results obtained from such experiments give little indication of how such systems would perform on a broader range of video content types, or indeed how differing content types can affect system performance. This lack of comprehensive evaluation means that the ability of many published techniques to accurately process large-scale amounts of real-world video is unknown.

This chapter discusses the video test suite employed in this research. We explain the composition of the test material, including format, size and content.

3.2 Aims of the test suite

One of the aims of this research has been to address the problem of inadequate video test suites for the evaluation of shot boundary detection systems. The focus of our research has been on the development of segmentation tools for use in broadcast digital video. Because of the size and diversity of the content types found in this domain, it was necessary to employ a substantial test suite to adequately evaluate the systems developed. In particular, and as part of a larger group of digital video researchers, we have developed a test suite that ia representative of the complexity of broadcast television. Towards this aim, it was deemed necessary for the test suite to have the following attributes:

1 The size of the test suite must be such as to provide a realistic evaluation of any system under test. Small test suites are unacceptable, as they provide no indication of how a system may be scaled-up to perform on large amounts of real-world video data.

2 The test suite must consist of numerous heterogeneous video content types. A test suite that does not include multiple content types is an inadequate representation of the complexity and diversity of broadcast digital video. To include, or even less to define, every conceivable content type found in broadcast video, is beyond the scope

of this research work. Nevertheless, a realistic effort must be made to incorporate as wide a range of diverse content types as possible.

Towards these aims, and in conjunction with other researchers working in the field of digital video, we developed the test suite described in this chapter.

3.3 Description of the test suite

The test suite employed consists of eight hours of broadcast television from a national TV station, comprising of all material broadcast from 1pm to 9pm on the 12th June 1998. The broadcast video was digitised in MPEG-1 format at a frame rate of 25 frames per second (total of 720,000 frames) and a resolution of 352*288 pixels (commonly known as the SIF standard). This was accomplished using a Pentium PC with a "Sphinx Pro" video capture board. For ease of manipulation, and to keep file sizes manageable, the video was digitised in 24 segments of 20 minutes each. Once captured, the video segments were transferred to a Sun Enterprise Server for further processing.

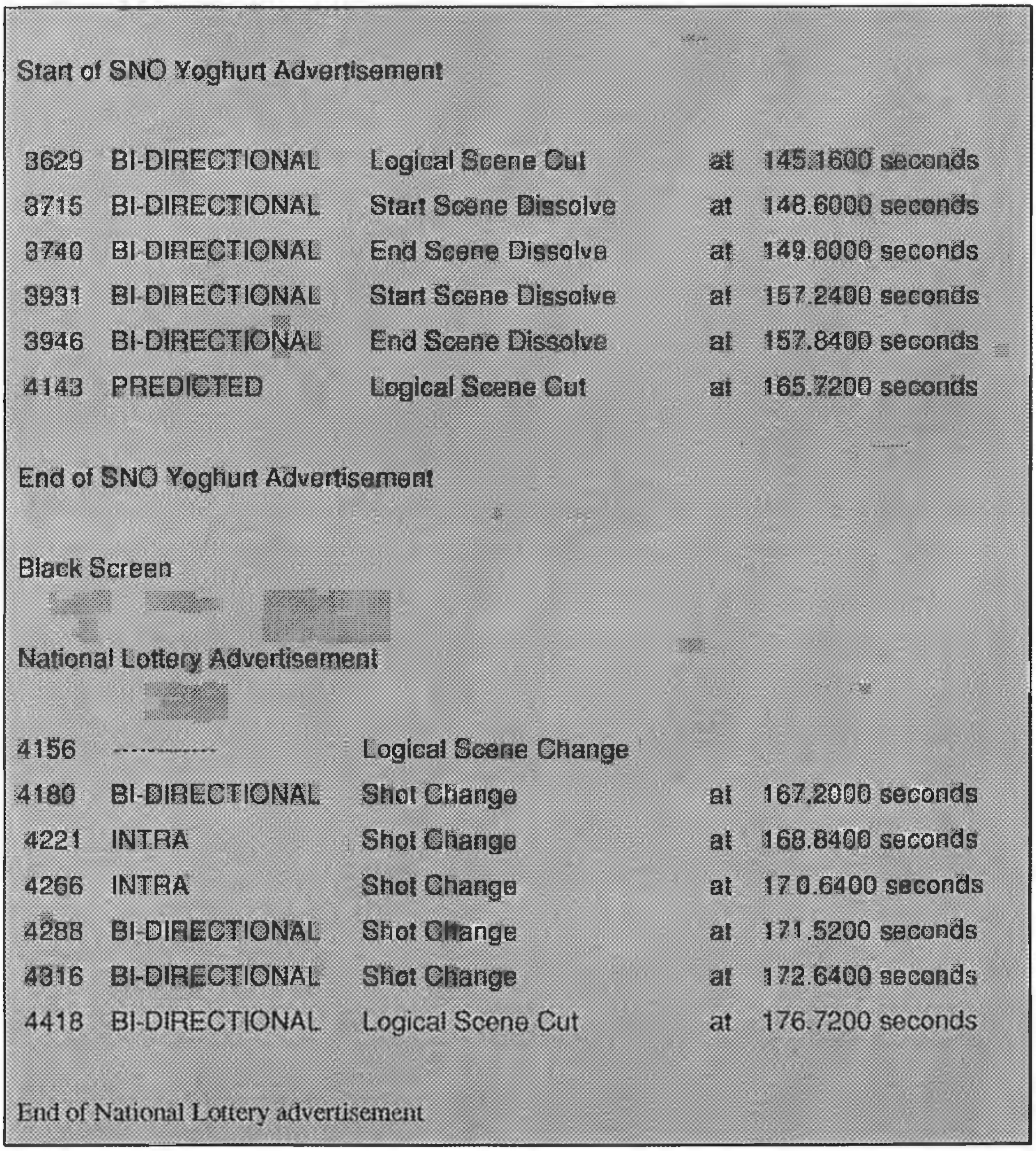
The test data incorporated a broad variety of program types, as well as a large number of commercials. Rather than sort the different content types into discrete test sets, the video was captured and stored "as is". This ensures that any given 20-minute segment may contain a variety of video content types. Thus the test set replicates the type of heterogeneous video most commonly seen on broadcast television.

To provide an authoritative guide to the test set, the locations and types of shot, scene, and program boundaries were manually analysed to give a series of detailed log files, each representing a 20-minute video segment. This collection of log files is referred to as the *baseline*, and represents a huge investment in time. The baseline allows us to compare the results generated by our detection algorithms to a 'ground truth'. It also enables us to calculate statistics such as the number of frames and shot boundaries found in each 20-minute segment, as well as in each content type. As noted above, the baseline contains extremely detailed semantic information. Although the work reported in this thesis focuses only on shot detection, the richness of the baseline will enable more complex methods and scene-based techniques to be evaluated successfully.

The baseline log files were generated by the author and two summer interns, working during 1998. Some checking across the log files was performed to ensure consistency. This task was accomplished using a simple markup logging tool developed by Aidan Totterdell, one of the summer interns. The overall task took eight person months of effort and was performed as part of our overall video project. An extract from one of the manually generated log files is shown below. Each shot boundary is logged in the following format :

```
Frame Number          MPEG Frame Type     Description              TimeStamp
```

Also listed is higher-level information about the shot, scene, and program boundaries. This information has been added to the baseline as freeform text by the manual indexers.



```
Start of SNO Yoghurt Advertisement

 3629  BI-DIRECTIONAL     Logical Scene Cut        at   145.1600 seconds
 3715  BI-DIRECTIONAL     Start Scene Dissolve     at   148.6000 seconds
 3740  BI-DIRECTIONAL     End Scene Dissolve       at   149.6000 seconds
 3931  BI-DIRECTIONAL     Start Scene Dissolve     at   157.2400 seconds
 3946  BI-DIRECTIONAL     End Scene Dissolve       at   157.8400 seconds
 4143  PREDICTED          Logical Scene Cut        at   165.7200 seconds

End of SNO Yoghurt Advertisement

Black Screen

National Lottery Advertisement

4156  -----------        Logical Scene Change
4180  BI-DIRECTIONAL     Shot Change              at   167.2000 seconds
 4221  INTRA              Shot Change              at   168.8400 seconds
 4266  INTRA              Shot Change              at   170.6400 seconds
 4288  BI-DIRECTIONAL     Shot Change              at   171.5200 seconds
 4316  BI-DIRECTIONAL     Shot Change              at   172.6400 seconds
 4418  BI-DIRECTIONAL     Logical Scene Cut        at   176.7200 seconds

End of National Lottery advertisement
```

A sample from one of the manually generated baseline files

The test suite contains eight broad content types. A brief explanation of each content type is given below:

1. News & weather: This content type includes two news broadcasts, one of 25 minutes and one of an hour. Also included was a 10-minute episode of Nuacht, the Irish language news. These video segments typify general news broadcasts, being a mix of anchorperson shots, split screen interviews, and outside broadcasts. Although this type of production does not typically include complicated camera techniques or visual effects, they do make limited use of features such as the aforementioned split screens. Also, outside broadcasts often exhibit a larger amount of global camera motion than in professional "fixed camera" studio productions. An obvious example of this is coverage of conflicts or civil unrest.

2. Soaps: Included are four complete episodes of soaps. They are "Home and Away", "Emmerdale", "Fair City", and "Shortland Street". Each episode was 30 minutes long. This is a very popular content type in general broadcast television, and one that can be characterised more easily than many other types. All soaps share the most critical constraint of the genre, that is, short production deadlines. This generally results in the fixed format of shots and scenes (camera positions rarely vary in such studio productions between episodes), and little post-production. Typically, the most difficult part of such productions to accurately segment are the opening and closing credits, as these often include more advanced camera techniques as well as a larger proportion of gradual shot transitions.

3. Cooking: This consisted of one half-hour cookery program. Surprisingly, this segment included many subtle gradual shot transitions. This is actually a common attribute of the genre, typically used to depict time passing as food is prepared or cooked. Such shot transitions are difficult to detect using colour techniques, as they tend to consist of (for example) a close-up shot of some dough before and after an ingredient such as sugar is added. The appearance of the new shot is generally not significantly different from the old, and the length of the gradual transitions (often greater than two seconds) only exacerbates the problem.

4. Magazine/Chat show: This was one 110-minute episode of a popular magazine show. Included are fitness, music, gardening, and film features, as well as interviews. This program contains a good mix of content types and shot transitions. This is an interesting content type, as although it is accorded a single category, it actually contains small

amounts of numerous other content types, such as those mentioned above. The bulk of such productions typically consist of interviews and discussions, composed of basic "talking head" and "around the table" shots with simple abrupt boundaries. However, they also include segments with high object motion (the archetypal "Mr. Motivator" fitness features), advanced computer effects (film reviews and MTV-style music videos), and numerous gradual transitions. Another common feature is the use of boom-cameras, used to sweep across the audience to focus on a particular individual. These shots naturally exhibit high global camera motion.

5. Quiz show: One half-hour episode of a popular local quiz show. This is perhaps the simplest content type to segment accurately, consisting for the most part of simple "talking head" shots interspersed with computer-generated displays (for scoreboards etc.). There tends to be little post-production of this content-type, leading to few complicated shot transitions.

6. Documentary: A short (20 minute) documentary charting the lives of some of the famous people of the 20th century. This content type provides very challenging material to segment accurately, due to the nature of the shot boundaries, and more importantly, the nature of the video itself. The documentary consists mainly of black and white footage from the 1920s-1940s, generally of extremely poor quality. In addition, the majority of shot boundaries are slow (2-3 second) dissolves and fades, which adds to the difficulty considerably.

7. Comedy/Drama: One full episode of "Touched by an Angel" (55 minutes) and one of "Keeping up Appearances" (35 minutes). The programs contained in this content type are somewhat similar to those from the "soaps" category. However, this content type differs in that the production schedule tends to be less frenetic, and production values higher, leading to more diverse and unusual productions. This is a broad content type, encompassing typical American–style "sit-coms" as well as made-for-television mini-series. Typically, the majority of the shot boundaries are still abrupt, but advanced camera techniques, unusual lighting effects, and computer-generated effects are somewhat common, especially in "X-files"/science fiction production types.

8. Commercials: Mixed among the above are a large number of commercials. As always, these provide varied and challenging material for segmentation. This is due to several reasons. Firstly, in comparison to regular programs, commercials typically have a huge number of shot transitions in a short space of time. Secondly, commercials frequently include much more advanced visual effects than programs, frequently using computer-

generated effects to distort, transform, and merge images. Finally, each commercial is designed to stand out and be different, in an attempt to arrest the interest of the consumer. This ensures that no two commercials are the same, and increases the difficulty of accurately detecting shot boundaries. Difficult as commercials may prove to be, however, no broadcast television test suite would be complete without them.

As realistic a representation of real broadcast video as the test suite is, it is nevertheless not a complete sample of the range of video content types present in that medium. The magnitude of the complete set means that an attempt to classify all possible content types would prove extremely difficult, and to obtain representative samples of each type would be impossible in the scope of this research. Notable absences from our test suite include sports (which is a huge content type, and would perhaps be better divided into sub-categories), Hollywood-style movies, and MTV-style music television. It is our hope that future work may help to address these shortcomings.

3.4 Analysis of the test suite

The baseline log files allow us to analyse the test suite to determine the number and type of shot boundaries in each 20-minute segment, and also in each of the eight content types mentioned above. This information can prove to be extremely useful in deciding why a particular segment, or content type, is proving difficult to segment accurately. Often, a high ratio of gradual transitions to abrupt transitions, or simply a large number of shot boundaries, can indicate an area of difficulty. Table 3-1 shows the test suite analysed by video content type, while table 3-2 shows the test suite analysed by video segment. Table 3-3 shows a detailed breakdown of each of the 24 video segments by content type, including the amount of each content type (in frames).

Of particular note in table 3-1 is the ratio of abrupt transitions (cuts) to gradual transitions in the different content types. This ranges from a very high ratio (67:1) for the "Quiz" content type, to a very low ratio (2:1) for the "Documentary" content type. As gradual transitions are generally more difficult to detect than cuts, this would indicate that the latter content type is the more challenging – which proves to be the case.

Video Type	# of Frames	# of Cuts	# of Gradual Transitions	Ratio of Cuts to Gradual Transitions
News and weather	134540	598	69	9:1
Soaps	144958	909	94	10:1
Cookery programs	37370	188	42	4:1
Magazine/chat shows	134985	759	64	12:1
Quiz shows	29093	269	4	67:1
Documentary	7494	47	23	2:1
Comedy/Drama	110618	839	72	12:1
Commercials	106976	1771	415	4:1
Total	**706034**	**5380**	**779**	**(average) : 15:1**

Table 3-1. Video test set analysed by video content type

Examining table 3-2, we would expect, based upon the ratio of cuts to gradual transitions, that segment 3 would prove quite challenging to segment accurately (ratio of 3:1). In the same way, we may speculate that segments 6 (69:1) and 23 (63:1) would prove considerably simpler. Again, in practice this proves to be the case, proving that the presence, or absence, of gradual transitions is a major factor in deciding the difficulty of analysing particular video clips. However, other factors, such as the type of camera effects used in the shots, and the amount of object and camera motion, can also have significant effects.

Video Segment	# of cuts	# of gradual transitions	Ratio	**Video Segment**	# of cuts	# of gradual transitions	Ratio
1	194	45	4:1	**13**	263	47	6:1
2	230	28	8:1	**14**	304	29	10:1
3	210	79	3:1	**15**	153	29	5:1
4	207	52	4:1	**16**	172	28	6:1
5	253	39	6:1	**17**	209	22	9:1
6	139	2	69:1	**18**	198	48	4:1
7	191	17	11:1	**19**	242	49	5:1
8	204	47	4:1	**20**	302	30	10:1
9	159	6	26:1	**21**	277	25	11:1
10	145	29	5:1	**22**	244	52	5:1
11	227	22	10:1	**23**	252	4	63:1
12	323	20	16:1	**24**	258	33	8:1
Note: Each segment is 30000 frames (20 minutes).							

Table 3-2. Video test set analysed by segment

Segment	Content Type	Duration (frames)	Segment	Content Type	Duration (frames)
1	News	22339	14	Soaps	26021
	Commercials	5020		Commercials	3979
	Soaps	2641	15	Soaps	3856
2	Soaps	26063		Commercials	3675
	Commercials	3937		News	22469
3	Soaps	4695	16	News	25920
	Commercials	7819		Commercials	4080
	Documentary	7596	17	News	25849
	Cooking	9787		Commercials	4151
4	Cooking	26583	18	News	4940
	Commercials	3417		Commercials	5999
5	Commercials	6411		Soaps	19061
	Magazine	23589	19	Commercials	9191
6	Magazine	30000		Soaps	18598
7	Magazine	25842		Comedy/Drama	2209
	Commercials	4158	20	Comedy/Drama	26089
8	Magazine	25303		Commercials	3911
9	Commercials	4697	21	Comedy/Drama	25814
	Magazine	30000		Commercials	4186
10	Magazine	20231	22	Comedy/Drama	23397
	Soaps	9769		Commercials	6603
11	Soaps	25299	23	Comedy/Drama	30000
	Commercials	4701	24	Comedy/Drama	2367
12	Soaps	2142		Commercials	8450
	Commercials	6777		News	19093
	Quiz	21081			
13	Quiz	8026			
	News	12569			
	Commercials	3452			
	Soaps	5953			

Table 3-3. Detailed breakdown of segment by video content type

3.5 MPEG-7 and the test suite

In October 1996, MPEG started a new work item to provide a solution to the problem of quickly and efficiently searching for various types of multimedia material. The new member of the MPEG family, called "Multimedia Content Description Interface" (in short 'MPEG-7'), will extend the limited capabilities of proprietary solutions in identifying multimedia content, notably by including more data types. In other words, MPEG-7 will specify a standard set of *descriptors* that can be used to describe various types of multimedia information. MPEG-7 will also standardise ways to define other descriptors as well as structures (*Description Schemes*) for the descriptors and their relationships. Finally, MPEG-7 will standardise a language to specify description schemes, i.e. a *Description Definition Language* (DDL).

This new standard does not specify how the extraction of such descriptors or features should be performed (for example, it does not specify whether such extraction should be manual or automatic). Nor does it specify the type of information retrieval system that can make use of the description. We can see then, that MPEG-7 is primarily concerned with the definition and standardisation of metadata for multimedia information. This information may include still images, 3D models, audio, speech, video, and also information about how these elements are combined in a multimedia presentation.

In MPEG-7 parlance, we can consider the elements of our baseline log files to be *descriptors* of our video test suite, as they describe the contents of the test suite. Further, the specification of the structure of the log files can be considered to be a *Description Scheme*, as it defines how the log files represent the information contained in the test suite.

At the time of writing, however, the syntax for MPEG-7 encoding is not defined, and will not be until the end of 1999. At that time it may be useful to transform our present baseline encoding into MPEG-7 format.

3.6 Summary

This chapter detailed the rationale behind the construction and analysis of the video test suite employed in our research. To successfully evaluate systems designed for broadcast video applications it is necessary to accurately simulate the large amount of diverse content types found in that media. Failure to do so results in incomplete results, which do not reflect the systems ability to scale up to real world applications involving such heterogeneous video.

The composition of the video test suite was then described. In particular, we presented brief explanations of the different video content types, focusing on the particular aspects that uniquely identify each one with regard to automatic shot boundary detection. We also noted the absence of several important content types from the test suite.

Finally, we presented details of the test suite analysed by segment and by content type, noting the different ratios of cuts to gradual transitions, and how these variations may affect the performance of automatic shot boundary detection systems.

The next chapter focuses on the first of the colour-based shot boundary detection systems developed and evaluated during our research.

4.0 A Shot-Boundary Detection System Based on Colour Histograms

4.1 Introduction

Our research has focused on the application of content-based colour similarity measures to detecting shot boundaries in broadcast video. Towards this aim, we have developed shot boundary detection systems utilising various colour-based techniques and then investigated the possibility of combining these techniques.

This chapter describes the first such system developed during our research, based on generation and comparison of colour histograms. Included in this chapter are results obtained by evaluating this system upon the video test suite described in chapter 3.

4.2 Colour histograms

4.2.1 Motivation

Following the work reported in [3] and [31], we chose colour histograms as the basis of our first shot boundary detection system. Histograms are far more resistant to object and global camera motion than simple pixel comparison techniques. At the same time, they also provide a high degree of discrimination – it is unusual for two dissimilar images to have similar histograms.

As detailed in chapter one, the MPEG-1 encoding standard utilises the YUV (or YCbCr) colour model. The Y component represents luminance, or brightness, and therefore ranges from black to white. The U and V components represent chroma, or colour. Therefore, a single pixel in a decoded MPEG-1 frame will have three associated values – one Y, one U, and one V. Typically, each of these values will range from 0-255 and will be encoded in a single byte.

4.2.2 Creation of colour histograms

We chose to create three 64-bin histograms for each frame, one histogram for each component of the YUV colour model. The number of bins comprising each histogram is an important choice, too many bins leads to over-sensitivity to object motion and small variations between frames, while too few leads to poor discriminatory performance due to lack of resolution. As mentioned above, the values of YUV are encoded as a single byte, giving a possible colour range from 0 to 255. Therefore, each histogram bin is four pixel values "wide", providing some degree of insulation from pixels drifting between colour bins, while at the same time allowing sufficient resolution to detect significant pixel differences. Figure 4-1 shows how the histogram bins are related to the image pixel values.

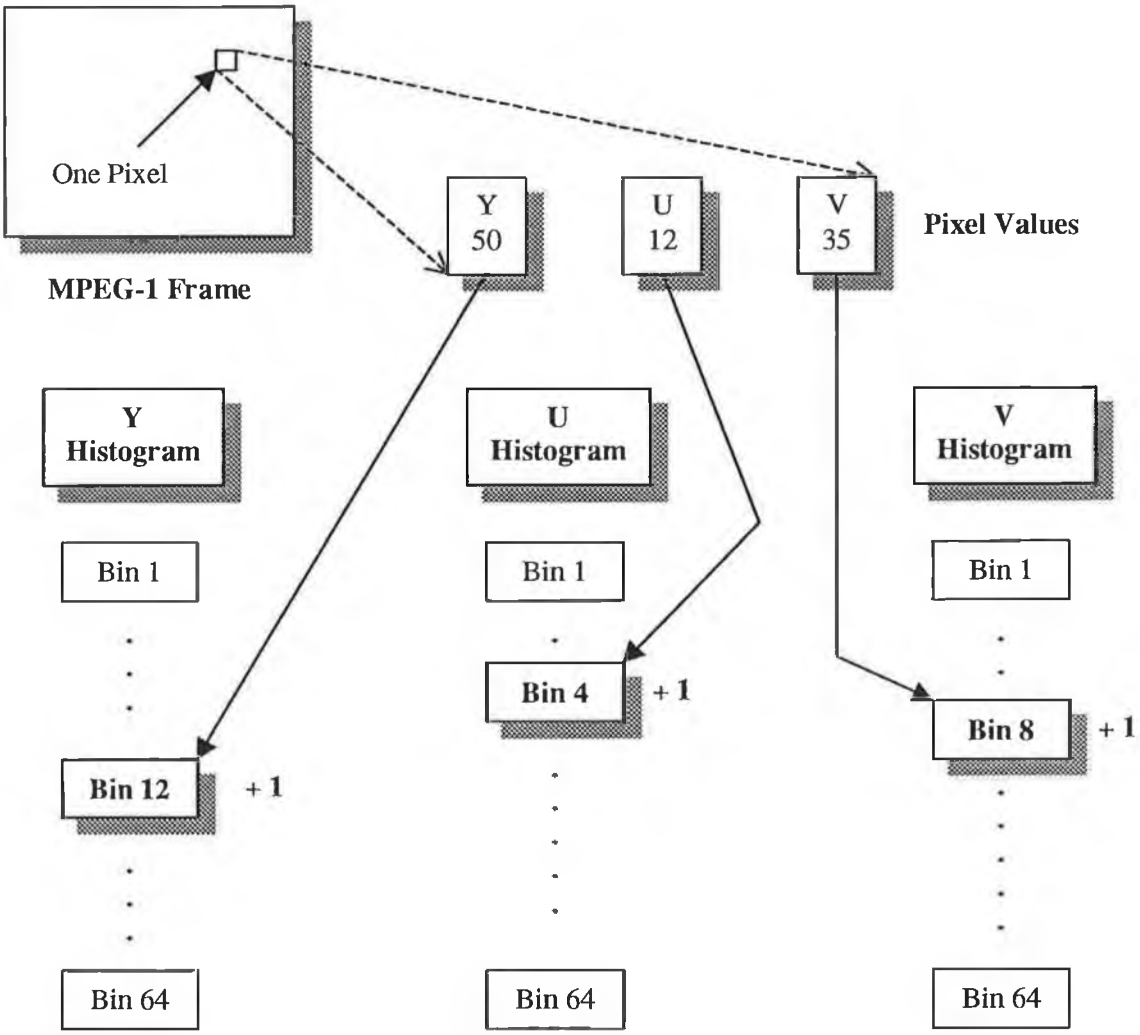


Figure 4-1. Histogram Generation

Once the three histograms have been created for a particular frame, they are then concatenated into a single 192-unit vector, which acts as the colour signature for that particular frame. As each frame represents only 1/25 of a second of video, the colour signature of adjacent frames should not vary by a significant amount, except when a shot boundary occurs. The next step in detecting shot boundaries, therefore, is to devise a measure for calculating the similarity between the colour signatures of adjacent frames.

4.2.3 The Cosine Similarity Measure

Many measures exist to calculate the histogram similarity, ranging from simple Euclidean distance to the more complex χ^2-test described in chapter 2. Following from the work reported in [3], we chose to employ the dissimilarity analogue of the Cosine Similarity Measure (CSM). This measure has been shown to outperform other, similar methods on small video test suites. We were interested to see how it would perform given a more diverse test collection.

We use the Cosine Similarity Measure to compare the colour histograms of adjacent frames. Given that the two 192-unit vectors, A and B, which represent the colour signatures of the two adjacent frames, The distance $D_{cos}(A,B)$ between vectors A and B is given by:

$$D_{\cos}(A,B)=1-\frac{\sum_{i=1}^{N}(a_i \cdot b_i)}{\sqrt{\sum_{i=1}^{N} a_i^2 \cdot \sum_{i=1}^{N} b_i^2}}$$

where a_i is one bin in A and b_i is the corresponding bin in B. As can be seen the cosine measure is related to the dot product of two unit vectors. The result is the cosine of the angle between the two vectors subtracted from one. Therefore a small value for D_{cos} indicates that the frames being considered are similar, while a large D_{cos} value indicates dissimilarity.

The CSM tends to identify clusters of false "shadow" shot boundaries around a real shot boundary. These false cuts tend to be at the immediately preceding or following frame and usually have a low, but sometimes significant, cosine value. To ensure that that these "clusters" of shot boundaries do not affect the overall results, the algorithm only declares the maximum cosine similarity value of the cluster to be the actual shot boundary. This simple technique works well, as broadcast television never includes several real shot boundaries within such a short space of time (each frame is displayed for only 1/25 of a second).

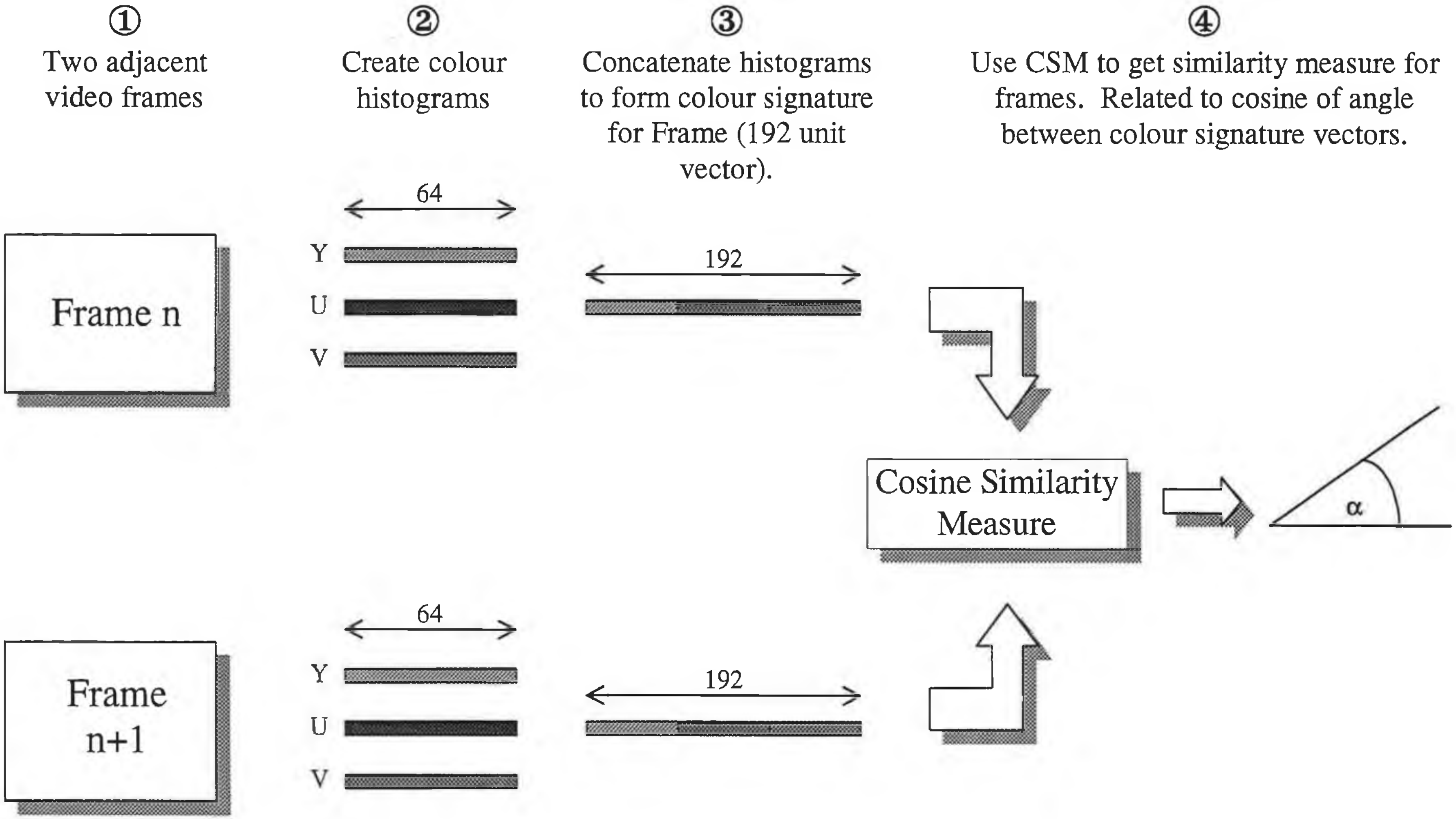

①
Two adjacent video frames
②
Create colour histograms
③
Concatenate histograms to form colour signature for Frame (192 unit vector).
④
Use CSM to get similarity measure for frames. Related to cosine of angle between colour signature vectors.
Frame n
64
Y
U
V
192
Cosine Similarity Measure
α
Frame n+1
64
Y
U
V
192

Figure 4-2. Overview of histogram shot boundary detection process

A high cosine distance measure can indicate one of two things. Firstly, it can (and should) signal that a shot boundary has occurred. Secondly, it could be the result of 'noise' in the video sequence, which may be caused by fast camera motion, a change in lighting conditions, computer-generated effects, or *anything that causes a perceptual change in the video sequence without being an actual shot boundary.*

Once the Cosine Similarity Measure has been applied to all the frames in the video sequence, we are left with a list of similarity values for each frame pair. We then chose a shot boundary detection *threshold* – cosine similarity values above this threshold are declared shot boundaries.

4.2 Results

4.3.1 Aims and methods

Before beginning the experiments proper, our segmentation algorithm was tuned on a number of small (5-10 minute) video segments extracted from the test suite. These training runs enabled us to determine useful threshold levels.

In reporting our experimental results, we use recall and precision to evaluate system performance. Recall is the proportion of shot boundaries correctly identified by the system to the total number of shot boundaries present. Precision is the proportion of correct shot boundaries identified by the system to the total number of shot boundaries identified by the system. We express recall and precision as:

$$\mathit{Recall} = \frac{\textit{Number of shot boundaries correctly identified by system}}{\textit{Total number of shot boundaries}} \qquad \mathit{Precision} = \frac{\textit{Number of shot boundaries correctly identified by system}}{\textit{Total number of shot boundaries identified by system}}$$

Ideally, both recall and precision should equal 1. This would indicate that we have identified all existing shot boundaries correctly, without identifying any false boundaries.

Although precision and recall are well established in traditional text-based information retrieval, there is as yet no standard measure for evaluating video retrieval systems. To present as clear a result as possible, we also use the E-measure, developed by van Rijsbergen [24]. This measure allows the researcher to associate a relative importance to both precision and recall. Given that *P* is precision, *R* is recall, and *b* is a weight of importance, the E-measure is defined as:

$$E = 1 - \frac{(1+b^2)PR}{b^2P + R}$$

For example, *b* levels of 0.5 indicate that precision is twice as important as recall, while a *b* level of 2 would indicate that recall is twice as important as precision. For our application, the segmenting of broadcast digital video, we believe that precision and recall are equally important. For this reason, we have chosen a value of 1 for *b*.

Recall, precision, and the E-measure are useful evaluation tools. However, by expressing results as a simple percentage they can give a misleading indication of system performance. For this reason we have chosen to include a summary of the actual figures obtained during the experiments. In reporting our results we chose a representative sample from the thresholds tested for inclusion in each graph. These samples include threshold levels that resulted in good results for all segments, and also samples from each extreme of the recall/precision spectrum.

Thresholds are not considered if they result in recall or precision figures of less that 0.5 for a majority of segments or content types. Although low recall or precision may be acceptable in some specialised applications, segmentation of large amounts of varied video requires reasonable levels to be useful.

In conducting the experiments we addressed specific questions with regard to shot boundary detection thresholds for broadcast video. We focused on the selection of correct thresholds for a mixture of video content types, as well as tailoring specific thresholds towards specific types. In particular, we were interested to see if pre-set, fixed thresholds were suitable for such a varied test set. The experiments conducted and results obtained are described below.

4.3.2 Do Fixed Threshold Values Perform Adequately on Video Containing Multiple Content Types?

To address the question of whether we can hard-code threshold values, we ran the algorithm, using a range of threshold values, on all 24 segments of the test set. A boundary detected by the algorithm was said to be correct if it was within one frame of a boundary listed in the baseline.

Recall and precision graphs are presented as figures 4-3 and 4-4 respectively. E-measure for the 24 video segments is presented in figure 4-5. A summary of results for the full video test set is also shown in table 4-1. The following points can be noted from these results:

1. On the middle threshold, the algorithm averages 85% recall, and 86% precision. However there is noticeable variation between the segments, as the algorithm performs better on different segments with different thresholds.

2. The algorithm performed poorly on segment 3 when compared to the rest of the test set. Even at the lowest threshold level, recall was only 75%, with a precision of 46%. This segment includes several commercial breaks. It also includes a lot of black and white footage from a documentary program. The colour-based method obviously has its discriminatory power reduced here, leading to poorer results.

3. The algorithm performed best on segment 6, typically achieving 98% precision and recall. This segment contains part of an episode of a magazine/chat show. Significantly there were no commercial breaks during this sequence. As commercials are generally the most difficult type of video to segment, this helps to explain the good results. Also, as noted in table 2, this segment has a huge ratio (69:1) of cuts to gradual transitions. The lack of difficult transitions makes for quite easy segmentation.

4. Lowering the threshold below a certain level does not guarantee better recall. Typically, once this level is reached (about 0.015 for our system), the increase in recall for a given threshold reduction is quite small, and is accompanied by a much larger loss of precision.

5. The opposite is also true, in that raising the threshold beyond a certain point gives decreasing precision results with rapidly falling recall.

6. For a majority (65%) of missed shot boundaries, examination of the raw data revealed a significant (>0.0075) cosine value for the appropriate frame pair. In these cases the fault of non-detection lies with the threshold selection, and not the detection ability of the algorithm itself. Attempting to employ a lower fixed threshold to detect these shot boundaries would result in a drastic decrease in precision. This suggests that an intelligent means of adaptive thresholding, perhaps using a known and reasonable threshold level as a starting point, could significantly improve upon the results obtained here. The need to improve methods of eliminating noise in the video stream is also a vital step if improvements are to be made in this area.

	Total # of shot boundaries	# correctly identified	# falsely identified	# missed	Recall	Precision	E-measure
Threshold 1 (0.010)	6159	5689	3775	470	92	60	78
Threshold 2 (0.020)		5472	1504	687	89	78	84
Threshold 3 (0.035)		5163	731	996	85	88	85
Threshold 4 (0.060)		4508	431	1651	74	92	82
Threshold 5 (0.15)		2789	195	3370	45	94	70

Table 4-1 – Total figures for the entire test set of 720000 frames.

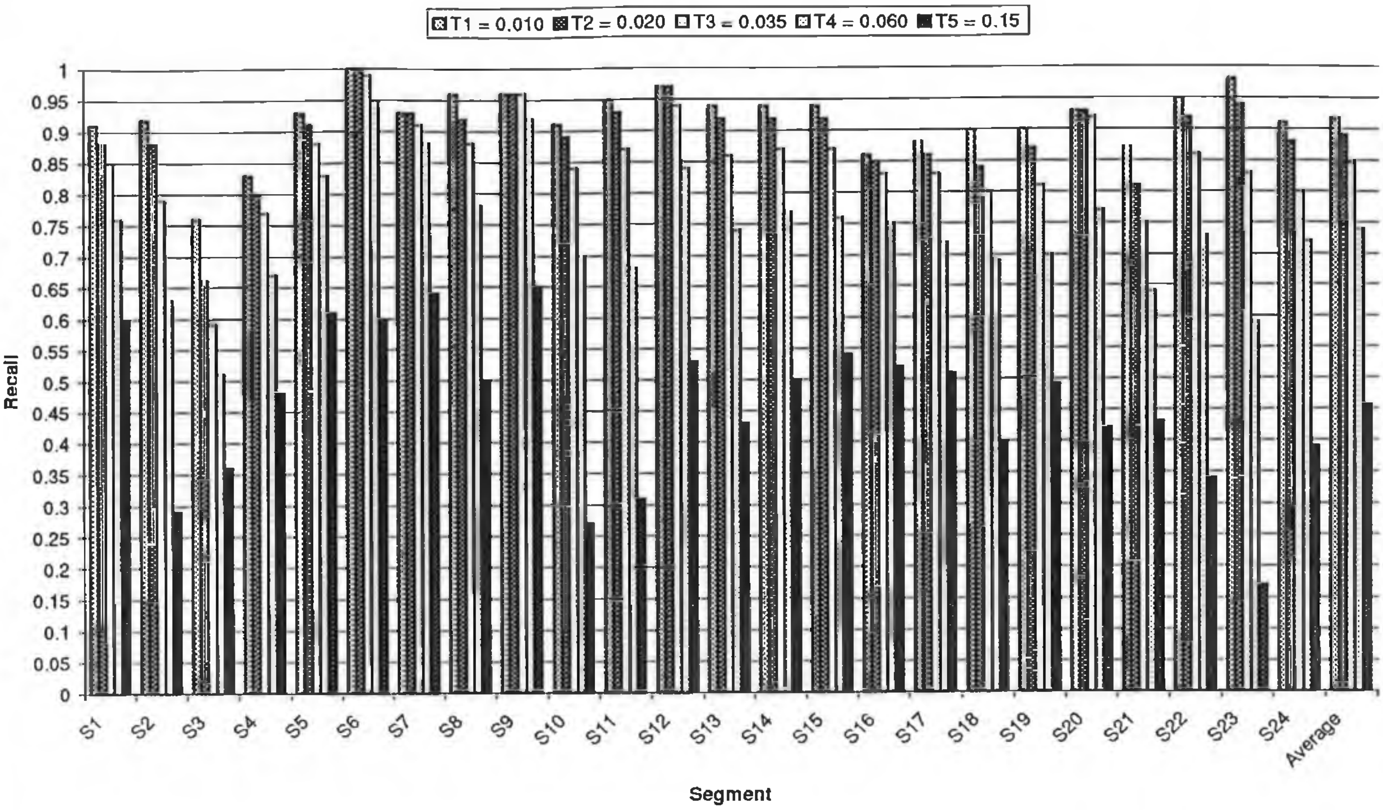


Figure 4-3: Recall for 24 video segments using 5 colour histogram thresholds

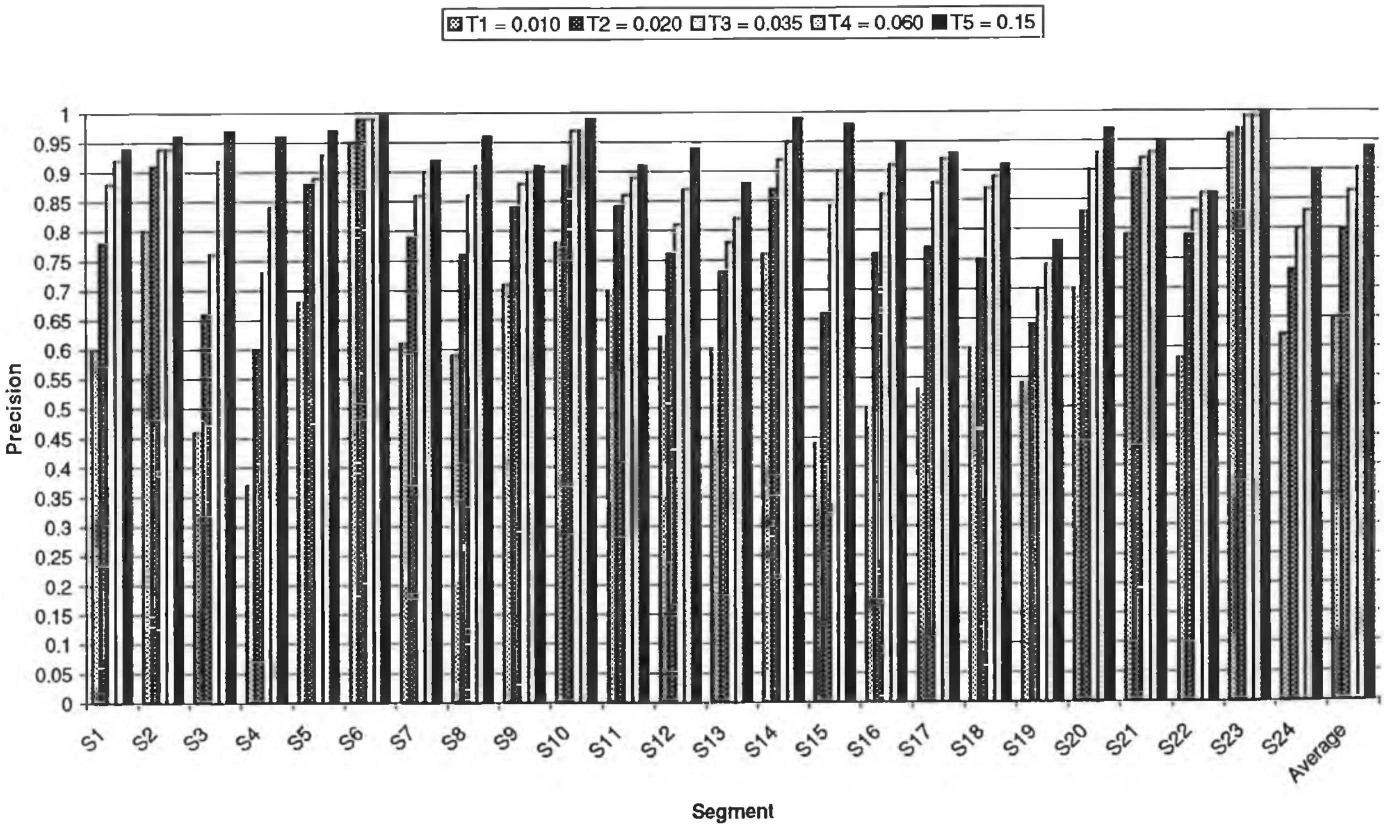


Figure 4-4: Precision for 24 video segments using 5 colour histogram thresholds

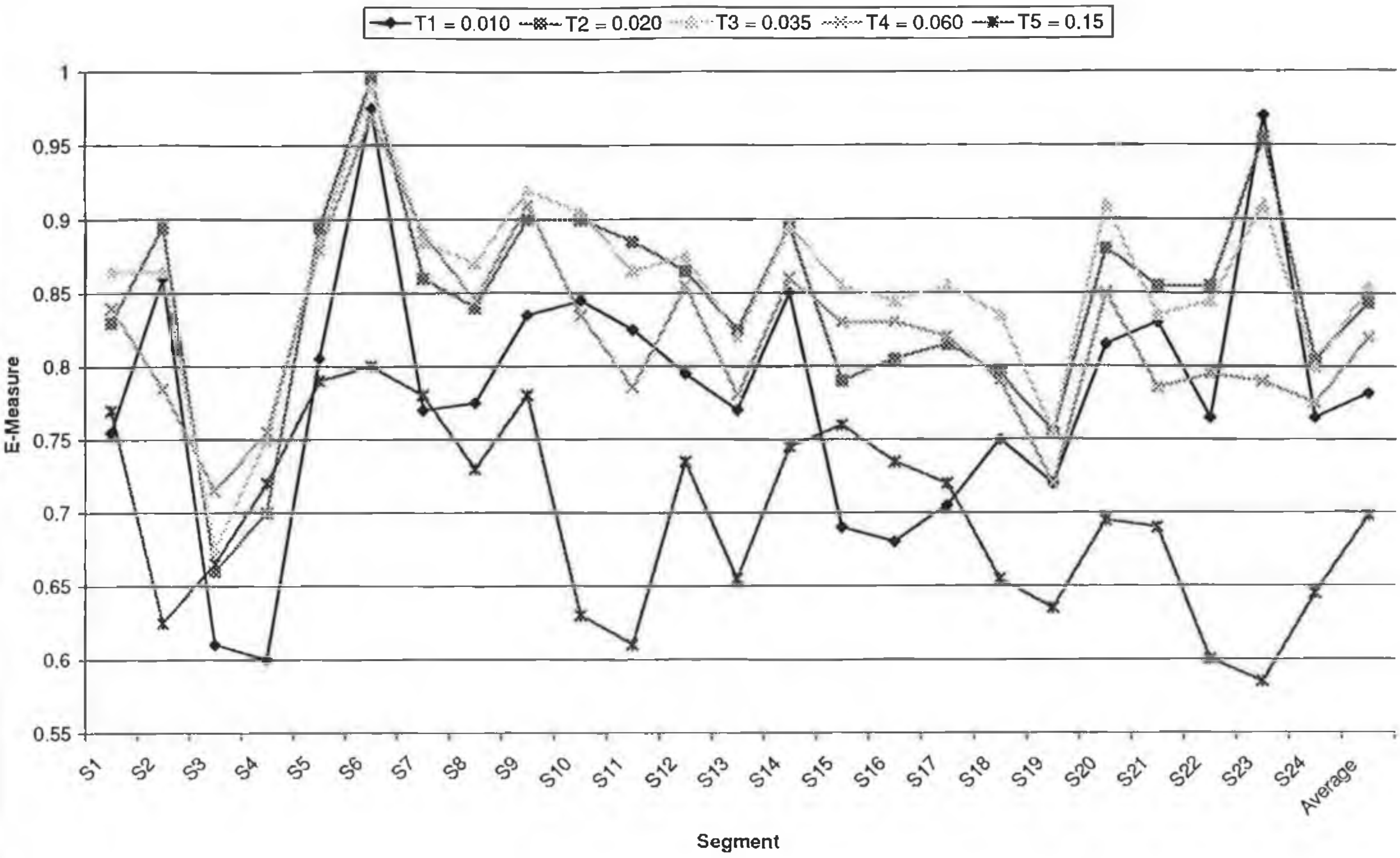


Figure 4-5: E-measure for 24 video segments using 5 colour histogram thresholds

4.3.3 Do Varied Video Content Types Affect the Results Obtained from Different Fixed Thresholds?

We have seen the results obtained by a selection of shot boundary detection thresholds on the 24 segments of the video test set. However, these results tell us little about why a particular segment/threshold combination is producing a particular result. Our second set of experiments explored how effective the system was at segmenting specific content types. This would show how different content type/threshold settings interacted and affected the overall result.

This second experiment requires that we examine the video test set by video content type, rather than by segment, as each segment contains a mix of content types. We employed the same five threshold settings as for section 4.3.1. Figures 4-6 and 4-7 show the recall and precision graphs for the eight video content types contained in the test set. Figure 4-8 shows the E-measure graph. The following general points can be noted:

1. Threshold levels can affect different video content types in markedly different ways. In some cases (for example between the "news" and "soaps" content types), the results are close enough to consider a single threshold value. However, the results for even these similar content types can vary by 20% for the same threshold.

2. In the case of dissimilar content types (commercials, documentary, cookery), the same threshold can produce completely different results. For example, a threshold of 0.035 results in 94% recall for the magazine/chat show content type, 79% for the commercials content type, and 9% for the documentary content type. Although this threshold setting performed best overall in section 5.1, these results show that it is totally inadequate for the mix of dissolves and black and white footage found in the documentary content type.

3. Again, examination of the missed shot boundaries revealed a majority (65%) that had significant cosine values. Had a reliable form of intelligent thresholding been employed in the algorithm, recall scores, which are currently quite poor, could be greatly improved.

We can also comment on the different video content types:

1. Commercials: This algorithm performed reasonably well when segmenting this content type, considering the complexity of some of the shot transitions present. Using a threshold setting of 0.035 (Threshold 3), 79% recall and 74% precision was achieved. However, moving to either end of the recall/precision spectrum quickly led to unbalanced results, which would prove unacceptable in our target application.

2. Soaps: This content type generally presented no difficulties to the system. On the middle threshold setting a precision of 92% was achieved. The low recall score of 76% was traced to the starting and ending credits of "Home and Away". This sequence contains some very difficult gradual transitions, which even our human baseline-creators found difficult to segment accurately.

3. News: As for soaps, the result for the news content type was generally good. Again, moving to extremes of the recall/precision spectrum led to poor results. When using a balanced threshold, recall and precision values averaged about 86%-87%.

4. Cookery: This content type proved difficult to accurately segment due to a large number of slow scene dissolves. Although low threshold settings (<0.030) afforded good recall, (85%-90%), the corresponding precision scores were poor (35%-50%). At the medium threshold settings (0.30-0.40) precision values are still quite poor (71%) although recall has improved to 83%. High thresholds, as expected, led to poor recall values (<50%). This content type demands an improved detection system before it can be segmented with confidence.

5. Magazine/chat show: Despite the varied content of this video type, the system performed quite well, probably due to the relative low ratio (12:1) of cuts to gradual transitions. Low and medium threshold values returned reasonable results with recall and precision ranging from 78%-98%. Higher threshold levels returned poor (<50%) recall scores, but gave little improvements in precision. This indicates that some proportion of the shot boundaries is being masked by noise in the video sequence.

6. Quiz show: The system performed well on this content type, which included few gradual shot boundaries. Low (<0.020) threshold values led to high (98%) recall scores with acceptable precision (78%). A more balanced threshold led to recall and precision scores of 97%. High threshold values are not suited to this content type, resulting in unacceptable (<55%) recall values.

7. Comedy/Drama: All threshold levels delivered good precision (>85%) on this content type, indicating that a high percentage of the shot boundaries are well defined. Recall dropped sharply from around 88% to 50% as the threshold was raised above 0.070, making such a setting unacceptable, even though doing so gave a precision of 100%.

8. Documentary: The system performed very poorly on this content type. This was due to the low ratio (2:1) of cuts to gradual shot boundaries and the large amounts of poor-quality black and white footage used as part of the documentary. At the medium threshold, which returned good results on all of the other content types, recall was only 9% and precision was 60%. In contrast to some of the other content types, best results were achieved with low (<0.015) threshold values, which gave around 64% recall and 52% precision. This content type highlights the difficulties of selecting one global threshold for broadcast video. Although the low scores achieved here were balanced in the overall system graphs (section 5.1) by the results obtained elsewhere, it is obvious that this content type demands more advanced shot boundary detection methods than our system currently offers.

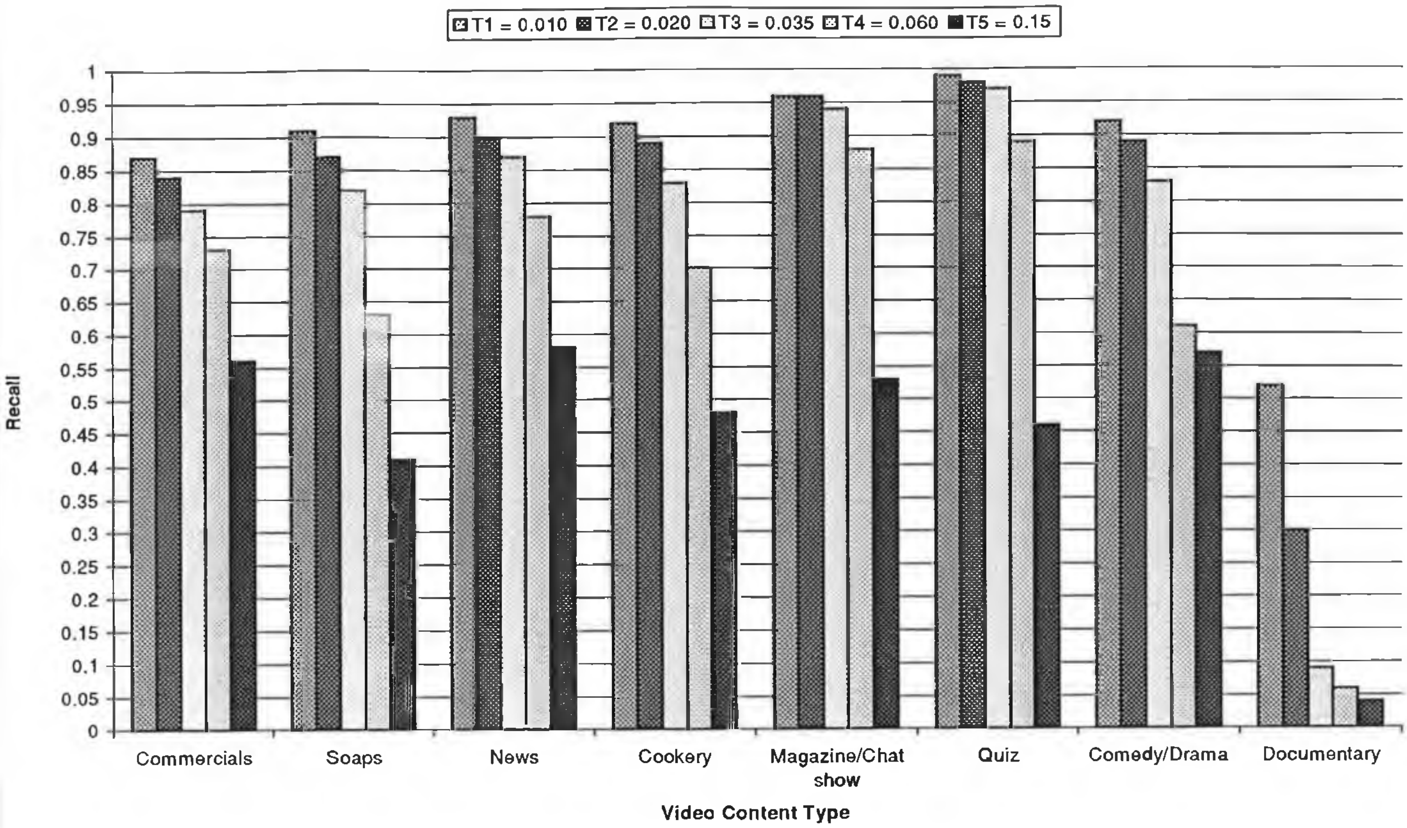


Figure 4-6: Recall for 8 video content types using 5 colour histogram thresholds

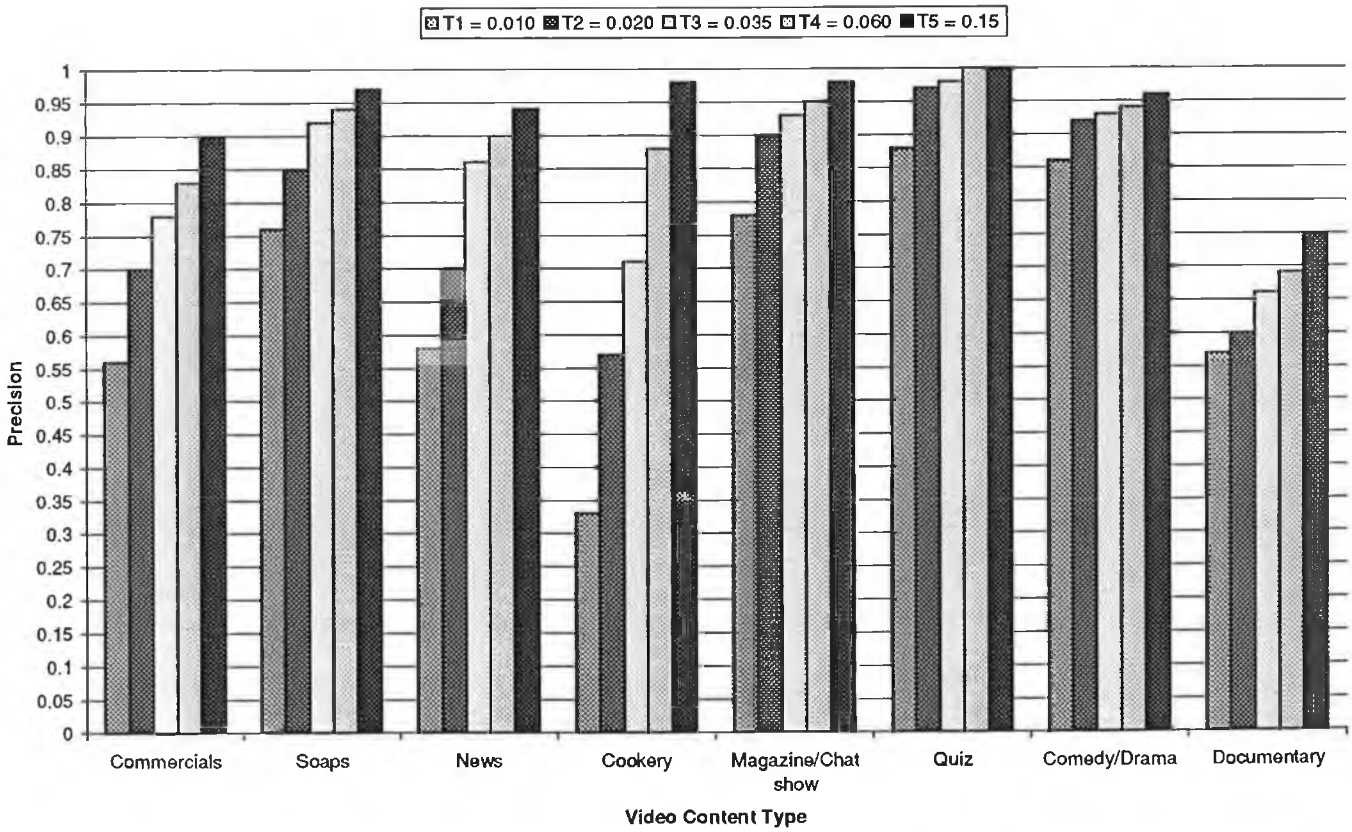


Figure 4-7: Precision for 8 video content types using 5 colour histogram thresholds

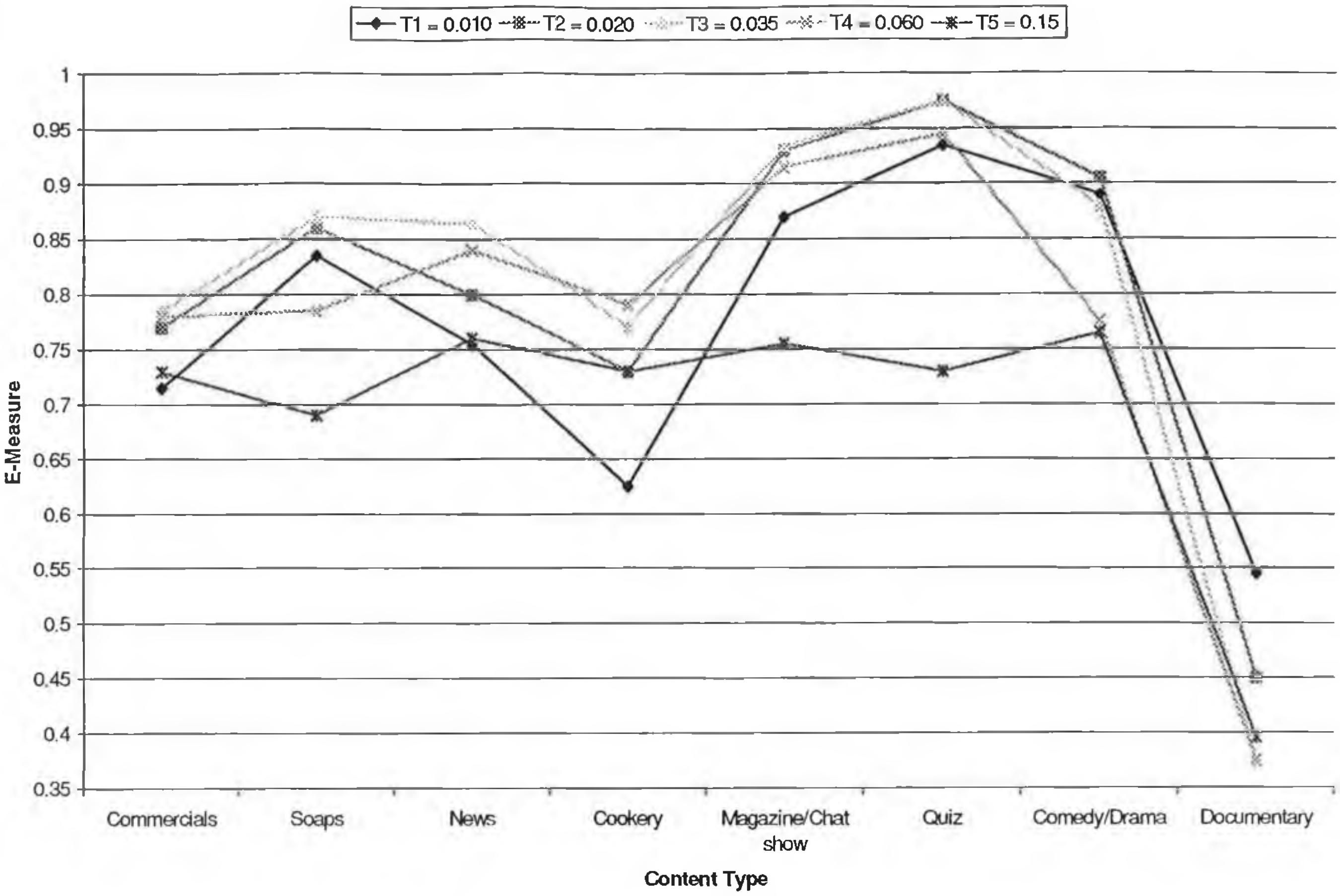


Figure 4-8: E-measure for 8 video content types using 5 colour histogram thresholds

4.4 Summary

This chapter described the first shot boundary detection system developed during our research, based on the generation and comparison of colour histograms. The rationale behind the selection of colour histograms was explained. The procedure for the generation and comparison of histograms was described, as well as the difficulties of selecting appropriate thresholds for a mixed-content test suite.

We introduced the concepts of precision and recall as measures of segmentation accuracy. These measures will also form the basis of the evaluation of later shot boundary detection techniques. We presented the results obtained by evaluating this system upon the video test suite described in chapter three, along with analysis of the systems performance when applied to single, and multiple content types. We believe these results to be very reasonable based on the complexity of the video test suite. We noted that the system performed markedly different when presented with different content types, and that different content types often required different threshold settings to accurately segment.

The next chapter focuses on the development of our second method of shot boundary detection, based on the statistical technique of colour moments. Full results will be presented for this next technique, following the format of this chapter.

5.0 A shot-boundary detection system based on colour moments

5.1 Introduction

Our research has focused on the application of content-based colour similarity measures to detecting shot boundaries in broadcast video. Towards this aim, we have developed shot boundary detection systems utilising various colour-based techniques and then investigated the possibility of combining these techniques.

This chapter describes the second such system developed during our research, based on computation and comparison of colour moments. Included in this chapter are results obtained by evaluating this system upon the video test suite described in chapter 3. The presentation of these results follows the format laid out in chapter 4.

5.2 Colour Moments

5.2.1 Motivation

Although histograms are the most popular technique for representing the colour composition of an image, statistical techniques have also been used with high success rates reported on small test suites – in fact some studies report superior performance than histogram-based techniques [1, 23]. We chose to implement a system using colour moments to represent the colour distribution of a frame. By implementing a system based on a different algorithm to our first, but still based on colour comparison, we aimed to identify the strengths and weaknesses of each method. Also, We were interested to see the correlation between the specific successes and failures of each algorithm.

Once again, the calculation of the colour moments is based upon the three colour components (Y, U, and V) which comprise each decoded MPEG-1 pixel.

5.2.2 Computation of colour moments

As a probability distribution is uniquely characterised by its moments, we can represent a colour distribution by its first three moments. Given that P_{ij} represents the i-th colour component of the j-th pixel of the decoded MPEG-1 frame, we can define the first three colour components as follows:

- The first order moment, μ_i, defines the average intensity of each of the three colour components, Y, U, and V. It is defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} P_{ij}$$

- The second order moment, σ_i, defines the variance of the intensity of the three colour components. It is defined as:

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - \mu_i)^2\right)^{\frac{1}{2}}$$

- The third order moment, s_i, defines the skewness of the intensity of the three colour components. It is defined as:

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - \mu_i)^3\right)^{\frac{1}{3}}$$

Given that our colour model consists of three colour components, it is obvious that the colour moments for each frame will consist of nine values. Each of the three components will be characterised by it's average intensity, variance, and skewness.

To reduce the time required to index a large video segment, the colour moment values may be generated from a subset of the total image pixels. Experimentation with pixel subsets of various sizes indicates that a large time saving can be made by employing sub-sampling, without a significant loss of accuracy. For the system reported here, we chose to generate the colour moment values from every fourth pixel in the x and y directions. As the original image resolution was 252*288 (SIF standard), this reduces the number of pixel values from 101376 to 6336, reducing both the time and memory requirements of the algorithm.

5.2.3 Comparison of colour moment values

Given two adjacent frames, each characterised by a number of colour moments, the distance between the two frames I and Q may be computed as follows:

$$D_{mom}(I,Q)=\sum_{i=1}^{r} w_{i1}\left(\left|\mu_i(I)-\mu_i(Q)\right|+w_{i2}\left|\sigma_i(I)-\sigma_i(Q)\right|+w_{i3}\left|s_i(I)-s_i(Q)\right|\right)$$

where r is the number of components present in the colour model, and w_{ij} $(1 \leq j \leq 3)$ the weight of the contributions of the different moments for each colour component. As we use only a small number of moments, it is possible that two perceptually dissimilar images may have similar statistical properties and therefore a low D_{mom} value. However, this is extremely unlikely in real-world operation.

For our experiments, we chose to weight the contribution of each moment equally. This gives the most balanced view of image similarity, without over-emphasising any specific moment. Figure 5-1 shows an overview of the colour moment shot boundary detection process.

5.2.4 Shot boundary detection for the colour-moment system

As with the system based on colour histograms described in chapter 4, detection of shot boundaries involves applying a threshold to a list of colour moment frame similarity values. Those values that exceed the specified threshold are declared shot boundaries. As with all threshold-based systems, the choice of the correct threshold for a given content-type is vital. However, this selection is made difficult given our mixed content-type video test suite.

The colour moment system does not allow as broad a range of usable thresholds as does the histogram-based system. Moving to outside a certain range of thresholds quickly leads to unacceptable performance. However, even within this workable range we find great variation as we examine the effects of different thresholds on different video segments and content types. In choosing our thresholds, we must attempt to balance the following two points:

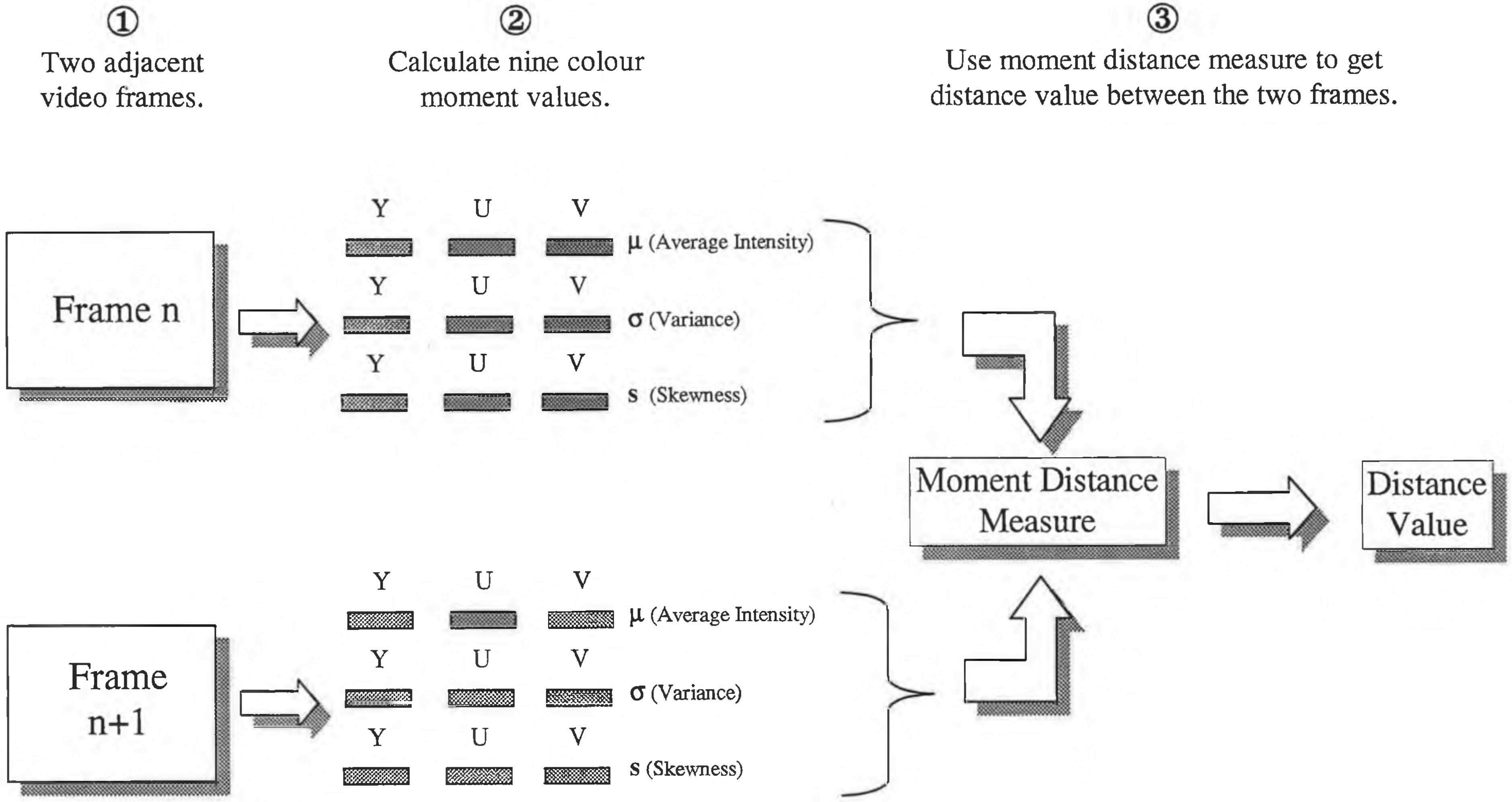


Figure 5-1. Overview of colour moment shot boundary detection process

- The need to prevent detection of false shot boundaries, by setting a sufficiently high threshold level so as to insulate the detector from noise.
- The need to detect subtle shot transitions such as dissolves, by making the detector sensitive enough to recognise gradual change.

Figure 5-2 shows a 700-frame sample from a segment of video. The content type shown is commercials, which is reflected in the large amounts of noise shown in the graph.

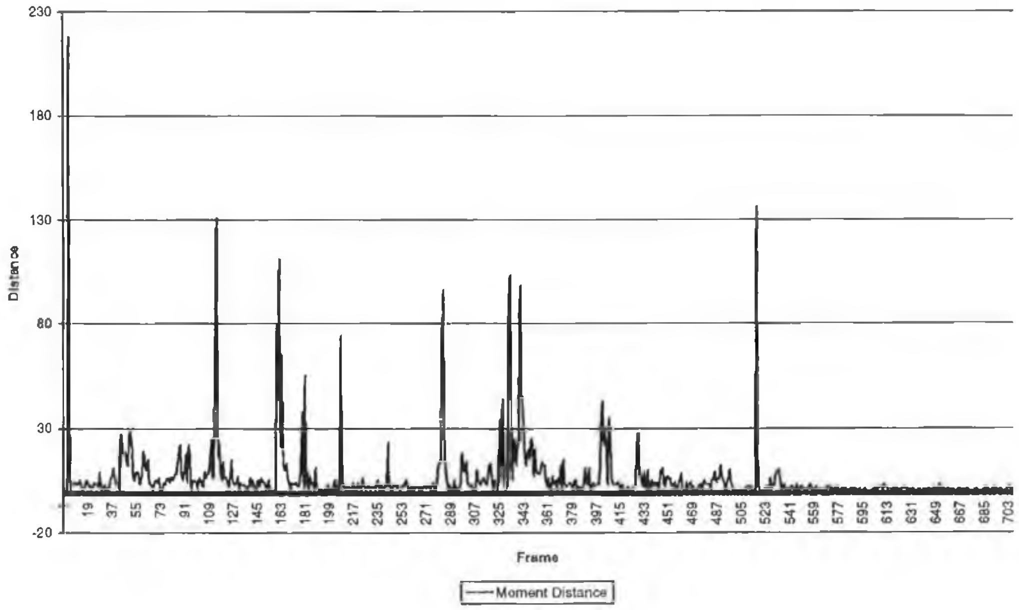


Figure 5-2: Colour moment values over a 700-frame video segment

Note that the large burst of activity around frame 341 is caused by fast object and camera motion, and not by the presence of a shot boundary. This sensitivity to perceptual changes, caused by elements other than shot boundaries, is the major weakness of this method. As can be seen from figure 5-2, such extreme bursts of activity are almost indistinguishable from real shot boundaries. Therefore they would almost certainly be declared as shot boundaries and the precision result for the algorithm would suffer as a result.

The bursts of activity from frame 41 to 101 are caused by two gradual transitions, the first stretching from frame 39 to 67, and the second from frame 78 to 101. These types of gradual transitions are characterised by such periods of low intensity. Such transitions are quite possible to detect reliably by the moment algorithm. Therefore the recall result of this algorithm is usually enhanced by identification of such shot boundaries.

5.3 Results

5.3.1 Aims and methods

This results section will closely mirror the format established in chapter 4. Certain sections will be omitted as they have already been covered in the last chapter. The evaluation metrics used will be briefly redefined, but the more general discussion as to the suitability of specific metrics has been excluded, to avoid repetition.

Before beginning the experiments proper, our segmentation algorithm was tuned on the same set of small (5-10 minute) video segments (extracted from the test suite) as were used for the histogram algorithm. These training runs enabled us to determine useful threshold levels.

In conducting the experiments we attempted to compare the results obtained with the colour moments algorithm to those discussed in the last chapter. In light of this, we again addressed specific questions with regard to shot boundary detection thresholds for broadcast video. We focused on the selection of correct thresholds for a mixture of video content types, as well as tailoring specific thresholds towards specific types. In particular, we were interested to see if pre-set, fixed thresholds were suitable for such a varied test set. The experiments conducted and results obtained are described below.

5.3.2 Do Fixed Threshold Values Perform Adequately on Video Containing Multiple Content Types?

To address the question of whether we can hard-code threshold values, we ran the colour moments algorithm, using a range of threshold values, on all 24 segments of the test set. A boundary detected by the algorithm was said to be correct if it was within one frame of a boundary listed in the baseline log files.

Recall and precision graphs are presented as figures 5-3 and 5-4 respectively. E-measure for the 24 video segments is presented in figure 5-5. A summary of results for the full video test set is also shown in table 5-1. The following points can be noted from these results:

1. The moment algorithm performs best within a relatively small range of shot boundary detection thresholds. Optimal settings to deliver balanced recall and precision for the majority of video types range from 7 to 20. Values outside this range degrade overall performance substantially.

2. Generally, as with all information retrieval systems, lowering the detection threshold improves recall (more shot boundaries are detected), at the expense of reducing precision (more false shot boundaries are detected). Selection of the optimum threshold, therefore, depends on the importance of correctly identifying the majority of shot boundaries, relative to the importance of avoiding detection of false shot boundaries. This is a design choice, and depends to a large extent on the target application of the system. For our experimental results, we define the optimal threshold for a particular method as the one that produces the best balance between precision and recall. For a general-purpose system such as ours, a large number of either false shot boundaries, or missed shot boundaries, is unacceptable. Applying this rationale to our experimental results, optimal thresholds are those in the range of 10 to 15.

3. Using these optimum thresholds, the algorithm averages 87% recall, and 80% precision. However there is noticeable variation between the segments, as the algorithm performs better on different segments with different thresholds. Compared to the histogram results, the moment algorithm exhibits a generally superior recall but inferior precision.

4. Once again, segment 3 proved to be challenging to segment accurately. However, the moment algorithm performed slightly better than the histogram algorithm in this case. Recall averages about 70% with a precision of 64%. This is in line with this methods general improvement of recall at the expense of precision.

5. Segment 6 produced the best results with recall and precision scores of 97%. This is inferior to those produced by the histogram algorithm, but not by a large margin. This result, along with the results from segment 3, indicate that the moments algorithm can detect the majority of abrupt cuts, but is significantly poorer at segmenting video with a large number of gradual transitions.

6. Analysis of the frames similarity values show that the colour moments measure is somewhat more sensitive than the histogram-based method, typically generating more significant distance measures for events such as fast object motion, global camera motion, and transitory changes in image brightness. However, this increased sensitivity also allows detection of more subtle shot transitions, which the histogram-based algorithm is typically insulated from.

	Total # of shot boundaries	# correctly identified	# falsely identified	# missed	Recall	Precision	E-measure
Threshold 1 (5)	6159	5789	4931	370	94	54	74
Threshold 2 (7)		5605	2518	554	91	69	80
Threshold 3 (10)		5358	1340	801	87	80	83
Threshold 4 (13)		5050	962	1109	82	84	83
Threshold 5 (15)		4866	727	1293	79	87	83
Threshold 6 (20)		4250	472	1909	69	90	80

Table 5-1 – Total figures for the entire test set of 720000 frames.

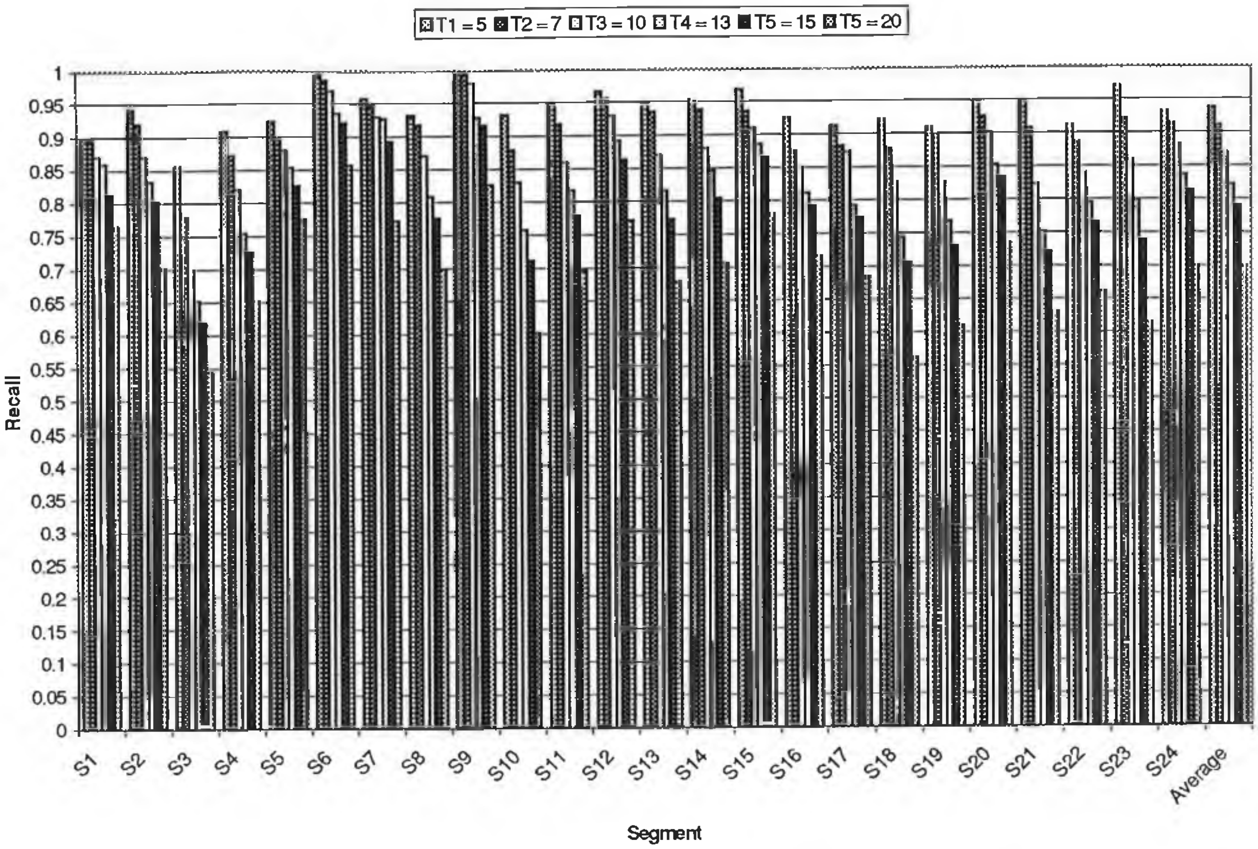


Figure 5-3: Recall for 24 video segments using 6 colour moment thresholds

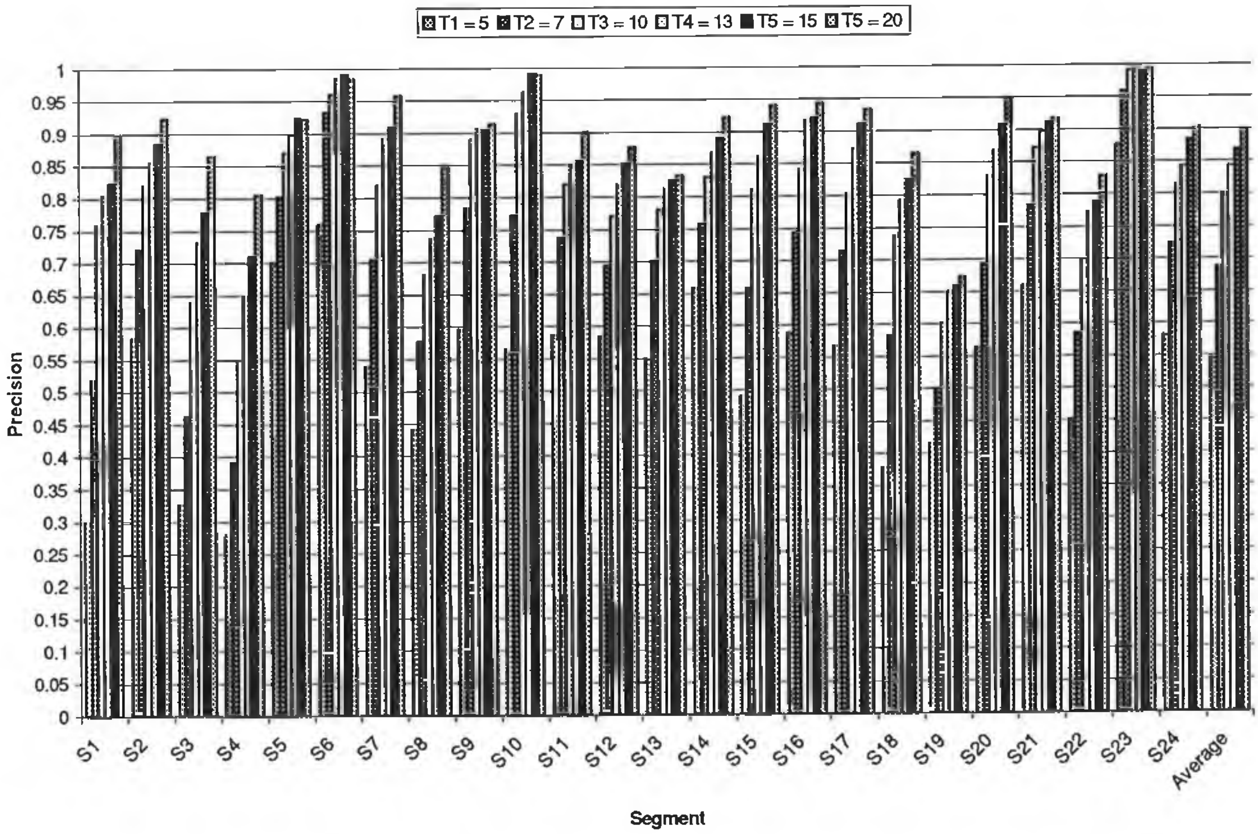


Figure 5-4: Precision for 24 video segments using 6 colour moment thresholds

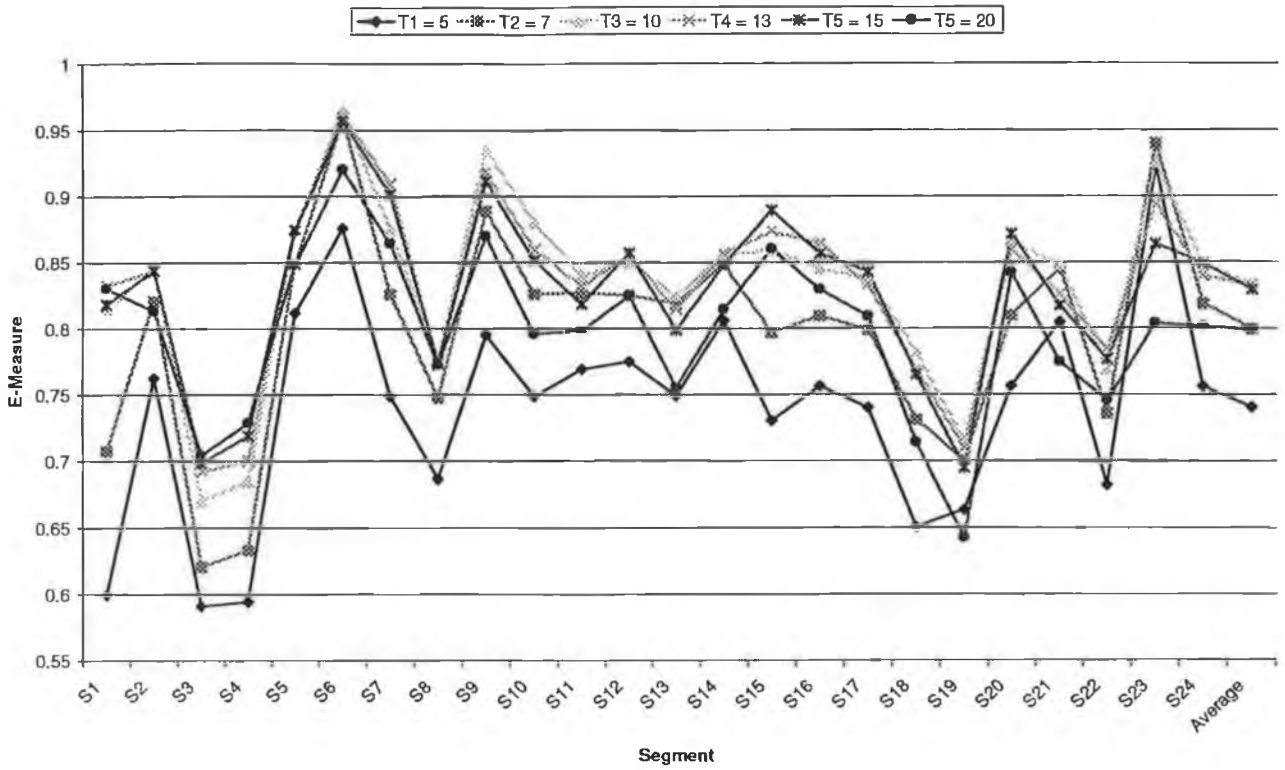


Figure 5-5: E-measure for 24 video segments using 6 colour moment thresholds

5.3.3 Do Varied Video Content Types Affect the Results Obtained from Different Fixed Thresholds?

We have seen the results obtained by a selection of shot boundary detection thresholds on the 24 segments of the video test set. However, these results tell us little about why a particular segment/threshold combination is producing a particular result. Our second set of experiments explored how effective the system was at segmenting specific content types. This would show how different content type/threshold settings interacted and affected the overall result.

This second experiment requires that we examine the video test set by video content type, rather than by segment, as each segment contains a mix of content types. We employed the same five threshold settings as for section 5.3.2. Figures 5-6 and 5-7 show the recall and precision graphs for the eight video content types contained in the test set. Figure 5-8 shows the E-measure graph. The following general points can be noted:

1. Threshold levels can affect different video content types in markedly different ways. In some cases (for example between the "news" and "soaps" content types), the results are close enough to consider a single threshold value. However, the results for even these

similar content types can vary by 15%-20% for the same threshold. In fact this effect is often extremely pronounced for the colour moment algorithm – in some cases changing a threshold by as little as 1 can result in markedly different results.

2. In the case of dissimilar content types (commercials, documentary, cookery), the same threshold produces more extreme differences. However, these variations are not as extreme as for the histogram based method. For example, a threshold of 10 results in 93% recall for the magazine/chat show content type, 85% for the commercials content type, and 60% for the documentary content type. This variation of results, although still very significant, is much less extreme than that reported in chapter 4.

3. When compared to the histogram based method, the colour moment algorithm performed best when applied to content types with a large number of subtle shot transitions. For example, applying the colour moment algorithm to the documentary content type produced precision and recall figures of 75% and 60% respectively. These results are a large improvement over the results of 52% and 57% produced by the histogram algorithm and reported in chapter 4.

We can also comment on the different video content types:

1. Commercials: The algorithm delivered good recall scores (82%-89%) at medium threshold levels (10-13), however precision was quite low, averaging at 72%. Thresholds in the low range of 0-10 deliver excellent recall (89%-95%), but at the expense of unacceptable precision (50%-60%). High threshold levels (>15) quickly lead to poor recall results, making them unsuitable for our purposes. The optimum threshold for this content type was 15, giving balanced precision and recall results of 80%.

2. Soaps: This content type is an exception to the typical behaviour of the colour moment algorithm, resulting in high precision values (91%) at the optimum threshold (13), but with mediocre recall (74%). Lower thresholds result in drastically poorer precision (58%-75%), which is not compensated by the increase in recall of 2% to 16%. Threshold values above the optimum result in steadily decreasing recall, for no gain in precision.

3. News: The algorithm was generally successful at segmenting this content type accurately, reflecting the absence of large numbers of gradual transitions and camera

effects. Threshold values from 0 to 9 are not suitable, due to poor precision, although they produce good recall results (90%-92%). Moving to a more balanced, medium threshold setting (13) improves precision greatly (to 89%) at the expense of a much smaller reduction in recall to 87%. Higher threshold levels would also be feasible if precision was a priority, as they achieve over 90% precision with reasonable recall scores of greater than 78%.

4. Cookery: This is still a challenging content type to segment accurately, in fact the colour moment algorithm performs worse than the histogram algorithm in this instance. Results are generally uninspiring, low thresholds give good recall (>87%) but poor precision (<50%). High thresholds produce the opposite, although precision remains poor until the threshold is set to above 19, at which point precision climbs to 85%, but at the expense of low (64%) recall. Our current colour moment algorithm is obviously unsuited to the mix of subtle shot boundaries, camera effects, and object motion found in this content type.

5. Magazine/chat show: Given the nature of this content type (few gradual transitions, little camera effects), the colour moments algorithm did not perform as well as expected. Although the results were certainly good, with a balanced threshold setting of 13 delivering 89% recall and 91% precision, this algorithm is quite sensitive to the small amounts of other content types present in this content type (see chapter 3 for a discussion of the characteristics of this content type). Raising the threshold beyond 15 produces slightly improved precision (94%), but at the expense of much poorer recall (77%). Threshold values below 9 greatly reduce precision, from 80% to approximately 65%, while raising recall by only 1%, to 97%.

6. Quiz: Again, the algorithm performed well on this content type, comprising of mostly abrupt shot boundaries with few sophisticated camera effects. A wide range of threshold values, from 7 to 20, produce good results, with recall and precision ranging from 90%-98%. Lowering and raising the threshold does not have a huge impact until extreme values are reached (less than 5 or greater than 20), at which point either recall or precision begins to suffer. Since the corresponding improvements in terms of recall and precision for choosing such an extreme threshold are minimal, it makes more sense to employ a balanced threshold in the 10-20 range.

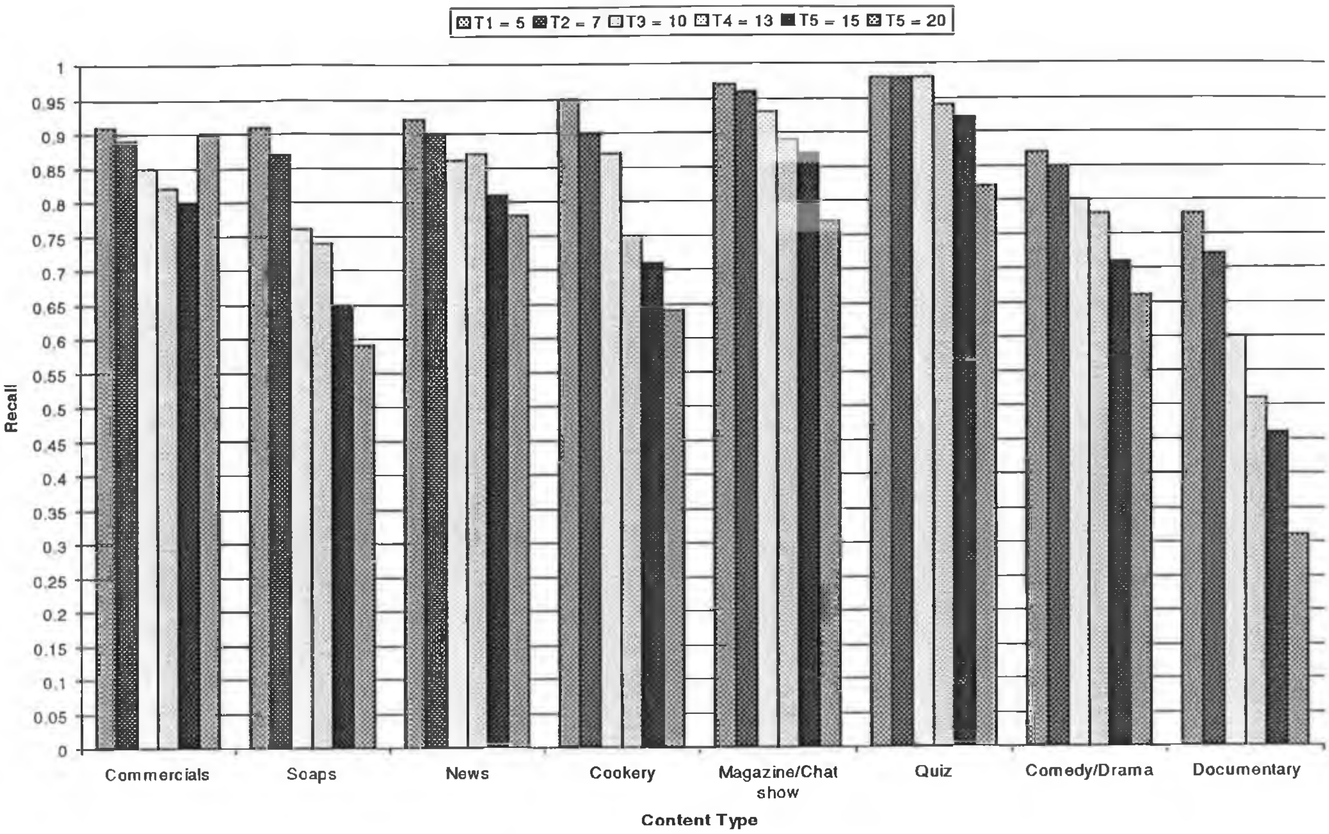


Figure 5-6: Recall for 8 video content types using 6 colour moment thresholds

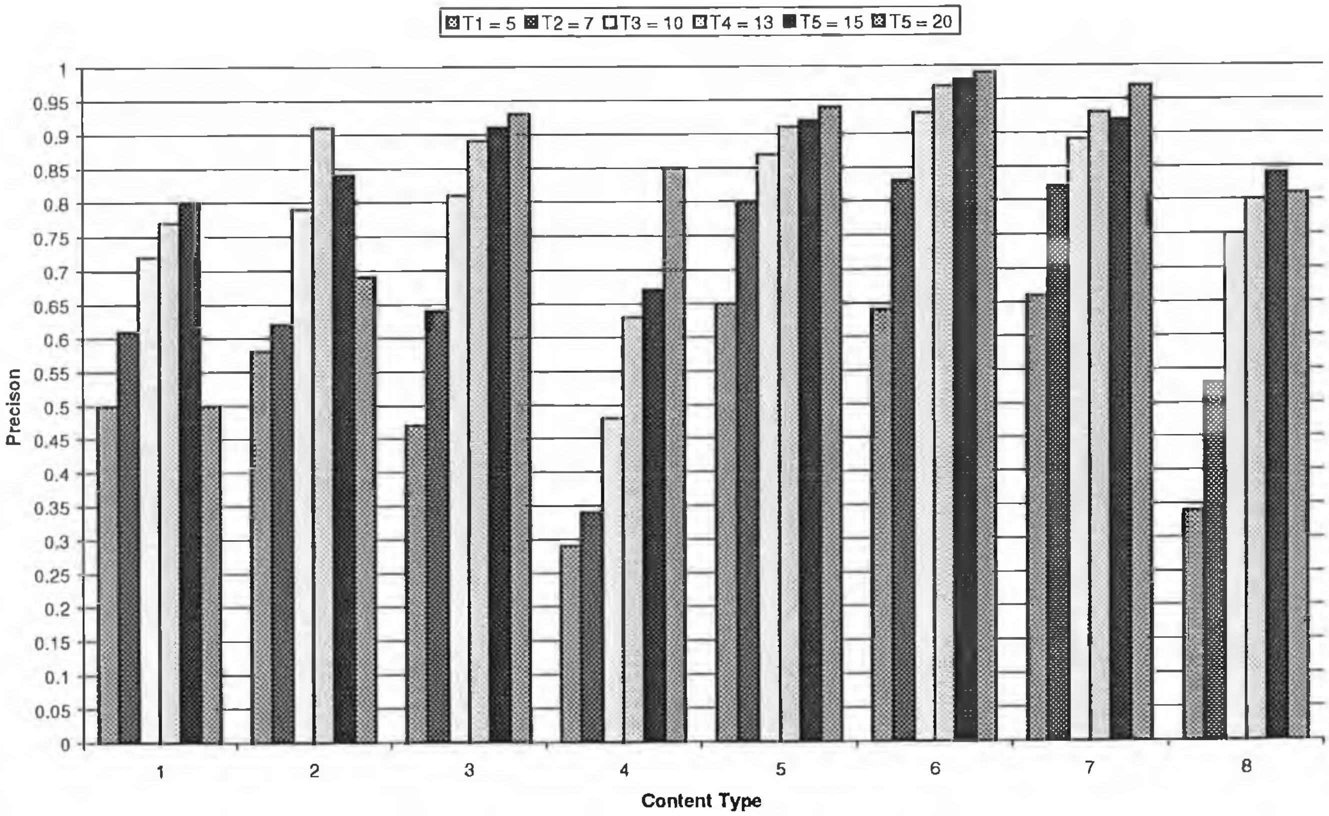


Figure 5-7:Precision for 8 video content types using 6 colour moment thresholds

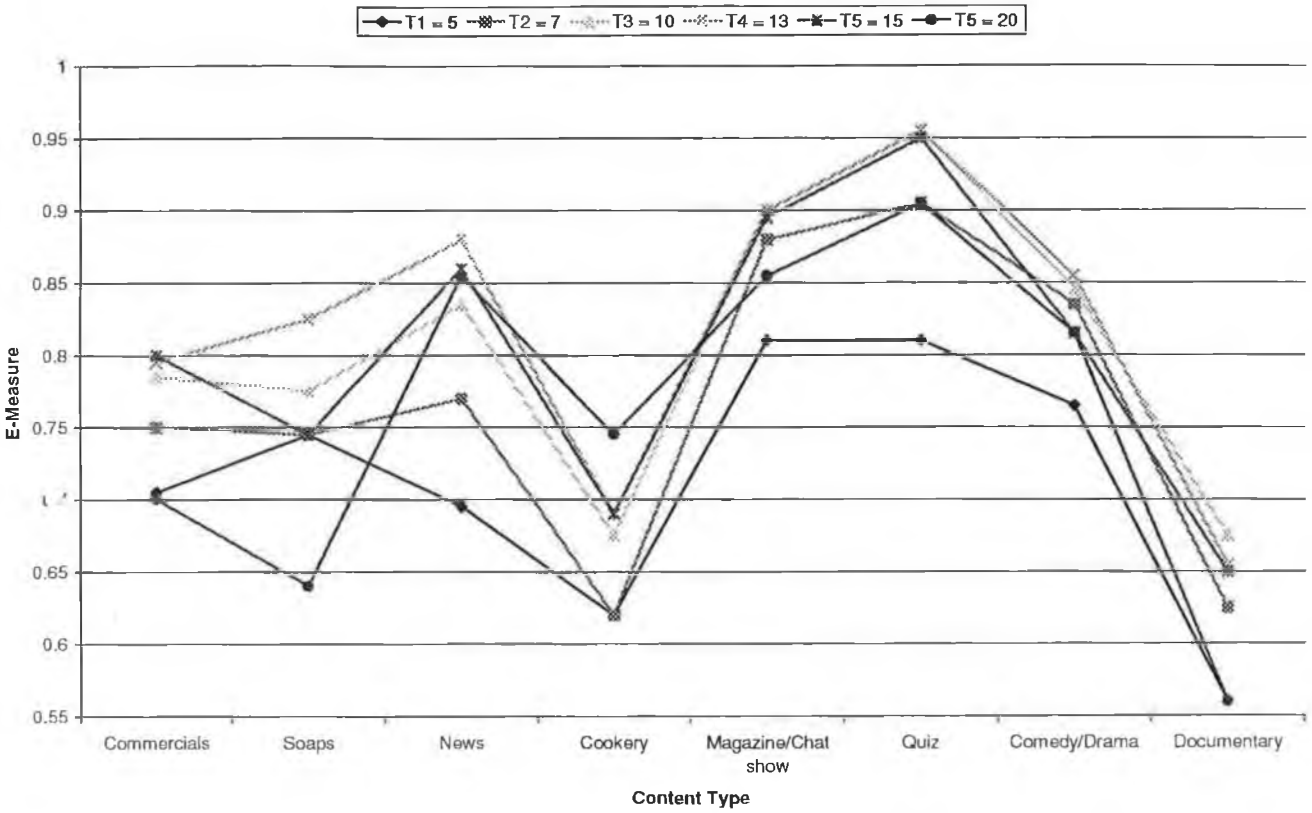


Figure 5-8:E-measure for 8 video content types using 6 colour moment thresholds

7. Comedy/Drama: This content type is a represents a middle ground of content types, in terms of types of shot boundaries present, post-production processing, and camera effects. As it comprises much of the bulk of normal television, results in this area are indicative of how we could expect the algorithm to perform on a wider video test suite. Typical results for medium threshold values (7-12) are recall in the range of 71%-85% and precision in the range of 82%-93%. Low threshold values reduce precision to 66%, too much to be acceptable for the marginal increase in recall. Threshold values approaching 20 however, may be worth considering for applications requiring high precision and non-critical recall, as they increase precision to 97%, while retaining a reasonable (66%) level of recall.

8. Documentary: The documentary content type is the most challenging in the test suite, as demonstrated by the poor results obtained by the histogram-based algorithm. The colour moment algorithm performs considerably better than its counterpart. However, the results obtained are still less than could be wished for. Threshold values below 8 deliver poor precision scores in the range of 30%-45%, and so are unacceptable, even though they result in a reasonable recall scores of 70%-78%. The optimum threshold

value is 10, which is slightly lower than for most other content types (The histogram-based algorithm also performs best with low threshold values, indicative of the number of extremely subtle shot boundaries present in this content type). Using this optimum threshold, recall is 60% and precision is 75%. These results are significantly superior to those obtained in chapter 4. High thresholds, in the range of 15-20, lead, as expected, to poor recall scores of between 30%-46%, and precision scores of 80%-84%.

5.4 Summary

This chapter described the second shot boundary detection system developed during our research, based on the computation and comparison of colour moments. The colour moments employed were defined, along with a distance measure to compare MPEG-1 frames based upon these moments.

In presenting our results, we once more employed precision, recall, and the E-measure as measures of segmentation accuracy. We presented the results obtained by evaluating this system upon the video test suite described in chapter three, along with analysis of the systems performance when applied to single, and multiple content types. We noted that the system performed markedly different when presented with different content types, and that different content types often required different threshold settings to accurately segment.

In presenting our results, we drew some comparisons between the performance of the colour moment algorithm when against that of the colour histogram algorithm. These comparisons will be explored more fully in chapter 7, which presents a more detailed comparison of the systems developed during our research.

The next chapter focuses on the development of our third method of shot boundary detection, based on combining the techniques of frame comparison based on histogram and moment similarity values.

6.0 A shot-boundary detection system based on a combination of colour histogram and colour moment techniques

6.1 Introduction

Our research has focused on the application of content-based colour similarity measures to detecting shot boundaries in broadcast video. We have developed shot boundary detection systems utilising various colour-based techniques and then investigated the possibility of combining these techniques.

This chapter describes the third system developed during our research, based on the combination of the systems described in chapters 4 and 5. As these techniques have already been comprehensively detailed, we shall focus on the rationale, method, and results of combining them, whilst referring the to the previously mentioned chapters for specific implementation details. Included in this chapter are results obtained by evaluating this system upon the video test suite described in chapter 3.

6.2 A combination system

6.2.1 Motivation

Having investigated two methods of shot boundary detection using colour similarities, and having studied other reported research on the subject, certain facts become apparent.

1. There is no "best" method of shot boundary detection method for all video types. Typically, different methods achieve different success rates depending on the type of video being processed. For example, colour-based comparisons may be quite capable of accurately segmenting the vast majority of abrupt shot boundaries, but they are less accurate when the video segment involved contains large numbers of gradual transitions

2. If the type of video to be processed is known in advance, it may be possible to employ a single optimum detection method, perhaps evaluated from small experimental runs. Of

course, the definition of ‘optimal’ will depend on the criteria being used: a typical application may require specifically high precision or recall levels, or there may be some time constraint that precludes the use of computationally-intensive methods.

3. However, given the almost infinite range of video content types found in broadcast television, it would appear that the “many heads are better than one” philosophy has proved most successful. This approach involves analysing the video segment using more than one detection technique, in an attempt to compensate for the weaknesses of one method with the strengths of another. Systems that employ this “combination” method include SWIM [32] and VideoQ [6]. Refer to chapter 2 for a more detailed description of these systems.

As our research has focused on colour-based techniques, we were interested to see if the histogram-based algorithm (chapter 4) and the moment-based algorithm (chapter 5) would produce similar results. Not only did we wish to compare simple precision and recall figures, we also wished to see the correlation between the specific shot boundaries detected by each system. Obviously, if the systems detect the same group of shot boundaries, and miss the same group, then combining them would produce little benefit. However, if the specific shot boundaries detected by each system are different, then a combined system would have the potential to improve the overall results.

6.2.2 Analysis of individual results

To investigate whether a combined system has the potential to improve upon the results of each individual system, we wish to examine the particular shot boundaries missed by each system. We can consider this situation using sets. Figure 6-1 shows the Venn diagram illustrating the set relationship. The elements of the venn diagram are detailed below.

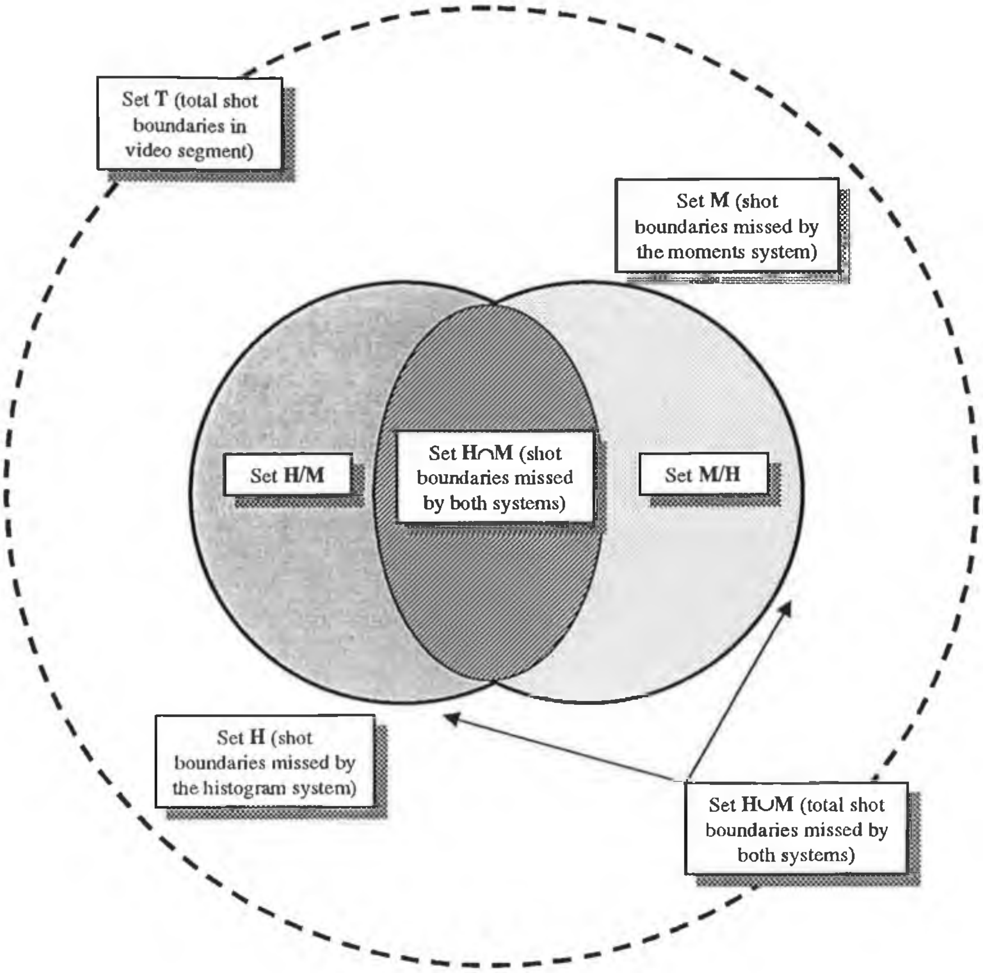

Set T (total shot boundaries in video segment)
Set M (shot boundaries missed by the moments system)
Set H/M
Set H∩M (shot boundaries missed by both systems)
Set M/H
Set H (shot boundaries missed by the histogram system)
Set H∪M (total shot boundaries missed by both systems)

Figure 6-1: Venn diagram for missed shot boundaries

- Set *T* is the total number of shot boundaries present in a particular segment of video.
- Set *H* is the set of shot boundaries not detected by the histogram-based system. *H* is a subset of *T*.
- Set *M* is the set of shot boundaries not detected by the moments-based system. *M* is a subset of *T*.
- Set *H∪M* is the total number of shot boundaries not detected by *either* of the two individual systems. *H∪M* is a subset of *T* and is composed of the subsets *H* and *M*.
- Set *H∩M* is the shot boundaries not detected by *both* of the two individual systems.
- Set *H/M* are the shot boundaries not detected by the histogram system, but which are detected by the moments system.
- Set *M/H* are the shot boundaries not detected by the moments system, but which are detected by the histogram system.

Figure 6-1 shows us the conditions that must exist if our combined system is to improve on the detection performance of the individual systems, namely:

- If the size of $H \cap M$, is large, compared to the size of the $H \cup M$, then the combined system has little scope for improving the overall results, as both individual systems are failing to detect the same set of shot boundaries.

- If the reverse is true, i.e. that $H \cap M$ is a small proportion of $H \cup M$, then the combined system could deliver a significant performance improvement over the individual systems. In other words, we wish the sets H/M and M/H to be as large as possible, so that each individual system is failing to detect different specific shot boundaries.

Having defined the optimal characteristics, table 6-1 lists the results of the above set analysis for the 24 video segments of the test suite.

Segment	$H \cup M$	$H \cap M$	Possible % improvement	Segment	$H \cup M$	$H \cap M$	Possible % improvement
1	44	29	34%	13	60	24	60%
2	63	25	60%	14	61	21	66%
3	137	79	42%	15	31	10	68%
4	70	40	43%	16	44	24	45%
5	47	25	47%	17	36	18	50%
6	5	0	100%	18	59	33	44%
7	25	9	64%	19	81	32	60%
8	43	19	56%	20	39	26	33%
9	11	2	82%	21	88	43	51%
10	36	20	44%	22	60	32	47%
11	48	21	56%	23	70	12	83%
12	34	14	59%	24	63	29	54%

Table 6-1 – Analysis of possible improvements using a combined system.

Recall that we are looking for a high percentage of shot boundaries that are in set H∪M but not in H∩M. Therefore, in table 6-1, we show the maximum possible percentage improvement that could be obtained, based on calculating the formula:

$$100\% - \frac{H \cap M}{H \cup M / 100}$$

The result is the percentage of currently undetected shot boundaries (those missed by both individual systems) that could be detected using a combination system.

As can be seen from table 6-1, the results strongly indicate that a combination system has the potential to greatly increase the recall of the current individual systems. The minimum possible improvement is 33%, in segment 20, while the maximum possible improvement is 100%, found in segment 6. Obviously however, any combination system will find it difficult to achieve this maximum increase in recall without a corresponding decrease in precision. Therefore, a suitable method of combining the results of the original systems must be employed, in order to maximise the benefits, and minimise the penalties.

6.2.3 Combination of results

Having shown that a combined system has the potential to perform better than its individual component systems, we now consider the question of how to combine the results from these systems. Our method of evidence combination is based on the following assumptions:

1. If either contributing system produces a very high distance measure for a particular frame pair, then a shot boundary exists at that position.

2. If both contributing systems produce moderately high distance measures for a particular frame pair, then a shot boundary exists at that position.

These assumptions can be expressed informally as "A shot boundary is declared when one system is extremely confident, or both systems are relatively confident, based upon their respective distance measures."

To implement this method of evidence combination, we require four shot boundary detection thresholds to be defined. Each of the two contributing systems has both an upper and lower threshold. The *upper* threshold is used to determine if the system is extremely confident of the presence of a shot boundary. The *lower* threshold is used to express relative confidence.

Given that $distance_a$ and $distance_b$ are the distance measures given by systems A and B for a particular frame pair. We can define $upper_a$ and $upper_a$ as the upper thresholds of systems A and B respectively, and $lower_a$ and $lower_b$ as the lower thresholds of these systems. The basic pseudocode implementation of the combined matching is then:

```
IF ( ($distance_a > upper_a$) OR ($distance_b > upper_b$) ) {
        Declare shot boundary at this position
}
ELSE IF ( ($distance_a > lower_a$) AND ($distance_b > lower_b$) ) {
        Declare shot boundary at this position
}
ELSE {
        No shot boundary exists
}
```

6.3 Results

6.3.1 Aims and methods

This results section will closely mirror the format established in chapters 4 and 5. Certain sections will be omitted as they have already been covered in the previous chapters.

Before beginning the experiments proper, our segmentation algorithm was tuned on the same set of small (5-10 minute) video segments (extracted from the test suite) as were used for the histogram algorithm. These training runs enabled us to determine useful threshold levels.

Once again, we use recall, precision, and the E-measure to evaluate system performance. Recall is the proportion of shot boundaries correctly identified by the system to the total number of shot boundaries present. Precision is the proportion of correct shot boundaries identified by the system to the total number of shot boundaries identified by the system. The E-measure is a weighted average of recall and precision. Refer to the results sections of chapters 4 and 5 for definitions of the above metrics.

We structured our experiments to address the same issues as have been raised in the previous chapter. Two experiments were carried out, the first focusing on the test suite by segment, and the second by content type. The results of these experiments are described below.

6.3.2 Do Varied Video Content Types Affect the Results Obtained from Different Fixed Thresholds?

To address the question of whether we can hard-code threshold values, we ran the combination algorithm, using a range of threshold values, on all 24 segments of the test set. A boundary detected by the algorithm was said to be correct if it was within one frame of a boundary listed in the baseline log files.

Recall and precision graphs are presented as figures 6-2 and 6-3 respectively. E-measure for the 24 video segments is presented in figure 6-4. A summary of results for the full video test set is also shown in table 6-2. The following points can be noted from these results:

1. Selection of suitable shot boundaries is made easier by reference to the results obtained in chapters 4 and 5. Each sample threshold setting on figures 6-3 to 6-5 refer to four values. For example, Threshold value 2 (T2 = 0.015, 0.035, 5, 20) indicates that:
    - The lower threshold for the histogram system was set to 0.015.
    - The upper threshold for the histogram system was set to 0.035.
    - The lower threshold for the moments system was set to 5.
    - The upper threshold for the moments system was set to 20.

	Total # of shot boundaries	# correctly identified	# falsely identified	# missed	Recall	Precision	E-measure
Threshold 1 (0. 010, 0.035, 5, 25)	6159	5605	1581	554	91	78	84
Threshold 2 (0.015, 0.035, 5, 20)		5543	1300	616	90	81	86
Threshold 3 (0.020, 0.040, 5, 25)		5481	1123	678	89	83	86
Threshold 4 (0.020, 0.050, 5, 30)		5481	1044	678	89	84	86
Threshold 5 (0.025, 0.040, 10, 25)		5358	872	801	87	86	86
Threshold 6 (0.030, 0.040, 10, 25)		5297	862	862	86	87	86

Table 6-2 – Total figures for the entire test set of 720000 frames.

2. The combination algorithm performs better than either the histogram or moment algorithms individually. At suitable threshold levels, it produces recall and precision figures of 89% and 85% respectively. This is compared to recall and precision results of 85% and 84% for the histogram method, and 87% and 80% for the moment algorithm. Although these differences may not seem large, they are in fact quite significant gains, especially considering the high recall and precision levels.

3. Once again, segment 3 proved to be challenging to segment accurately. However, the combination algorithm performed, on average, better than the previous algorithms. Recall averages about 68% with a precision of 69%. Although the recall score is 1% less than that achieved by the moment algorithm, the precision score is 5% greater. Recall is greatly superior (68% vs. 56%) to that achieved by the histogram algorithm.

4. Segment 6 produced the best results with recall and precision scores of 100%. This is slightly superior to both previous algorithms, which achieved scores of 93%-98%. Again this small increase is significant at such high recall and precision levels.

5. The results mentioned above confirm that combining individual algorithms can overcome the weaknesses of each. The combination algorithm is not superior in every respect to the systems described in chapters 4 and 5. In certain cases it may produce, for example, inferior recall results, as shown in point 3. However, the combination algorithm does produce more *balanced* results, meaning that achieving a relatively high recall score does not come at the expense of a poor precision result. As discussed in the results section of chapter 5, we believe that a balanced system is preferable when attempting to accurately segment large amounts of heterogeneous broadcast video.

6. The range of thresholds reported varies less than for the previous two chapters. This is due to the fact that having the results of previous experiments on the component systems available allows us to cut out a whole range of thresholds that are known to result in poor recall and precision scores. The resulting, tighter ranges of threshold values used in the graphs are a good representative sample of the results obtained.

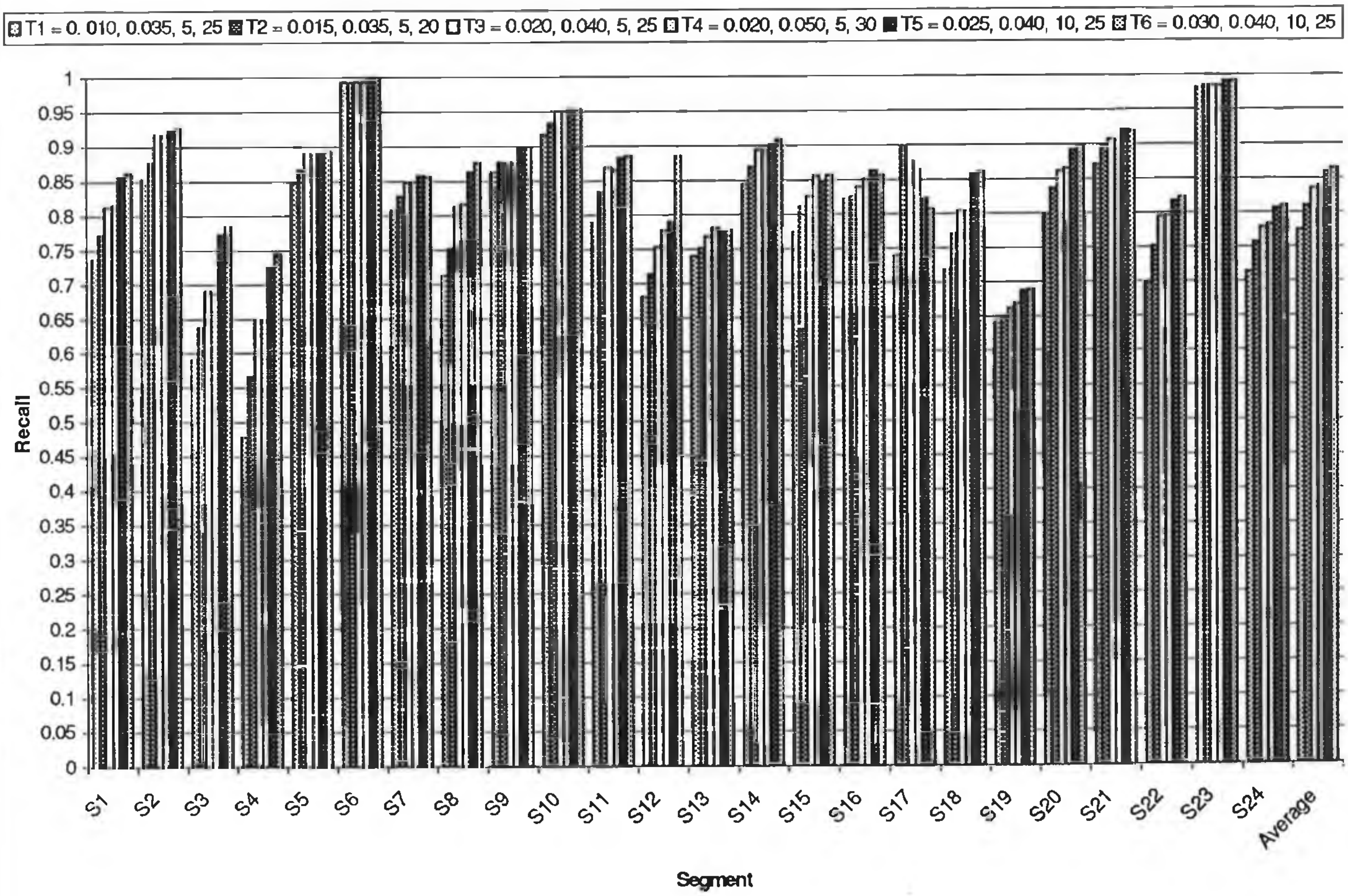


Figure 6-2: Recall for 24 video segments using 6 combination thresholds

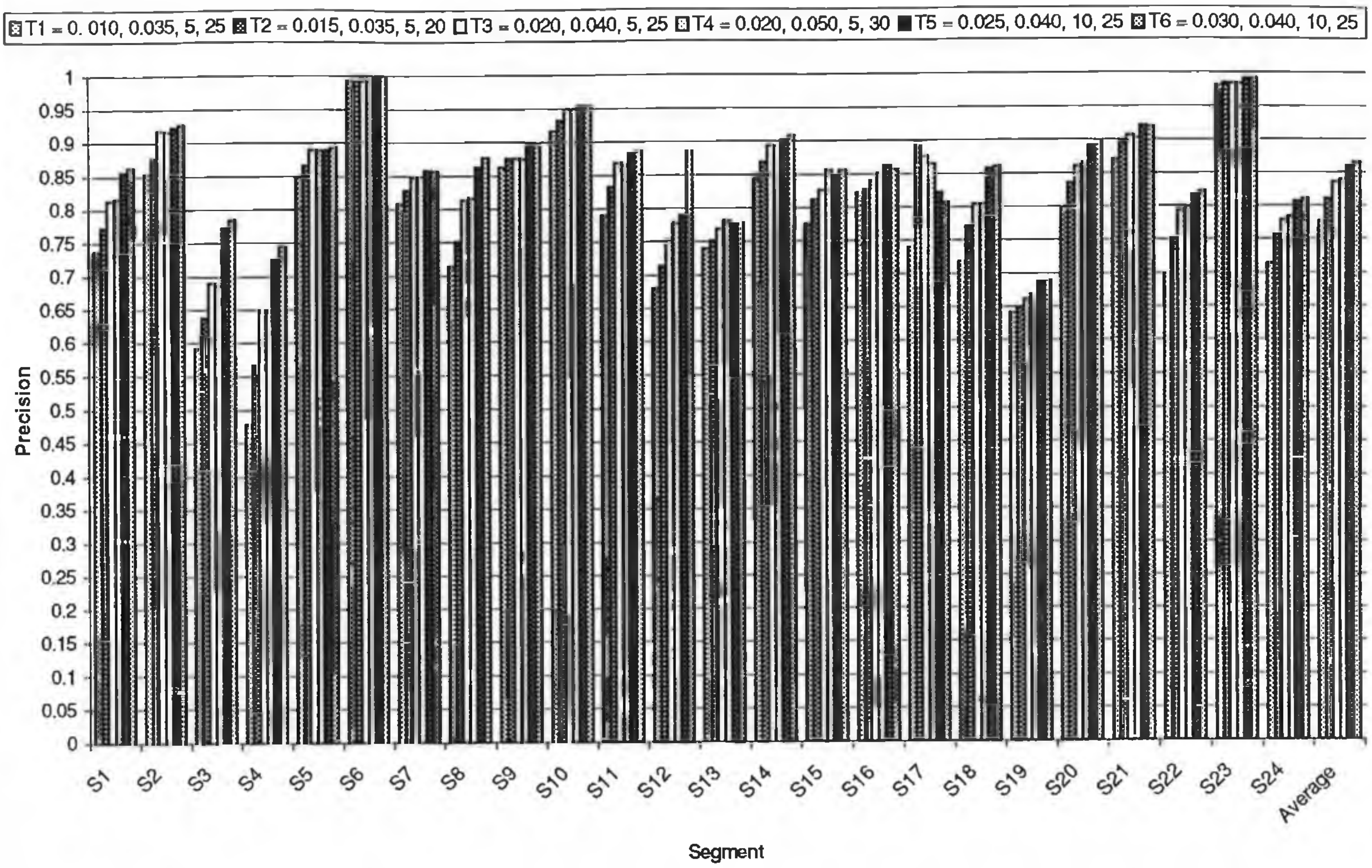


Figure 6-3: Precision for 24 video segments using 6 combination thresholds

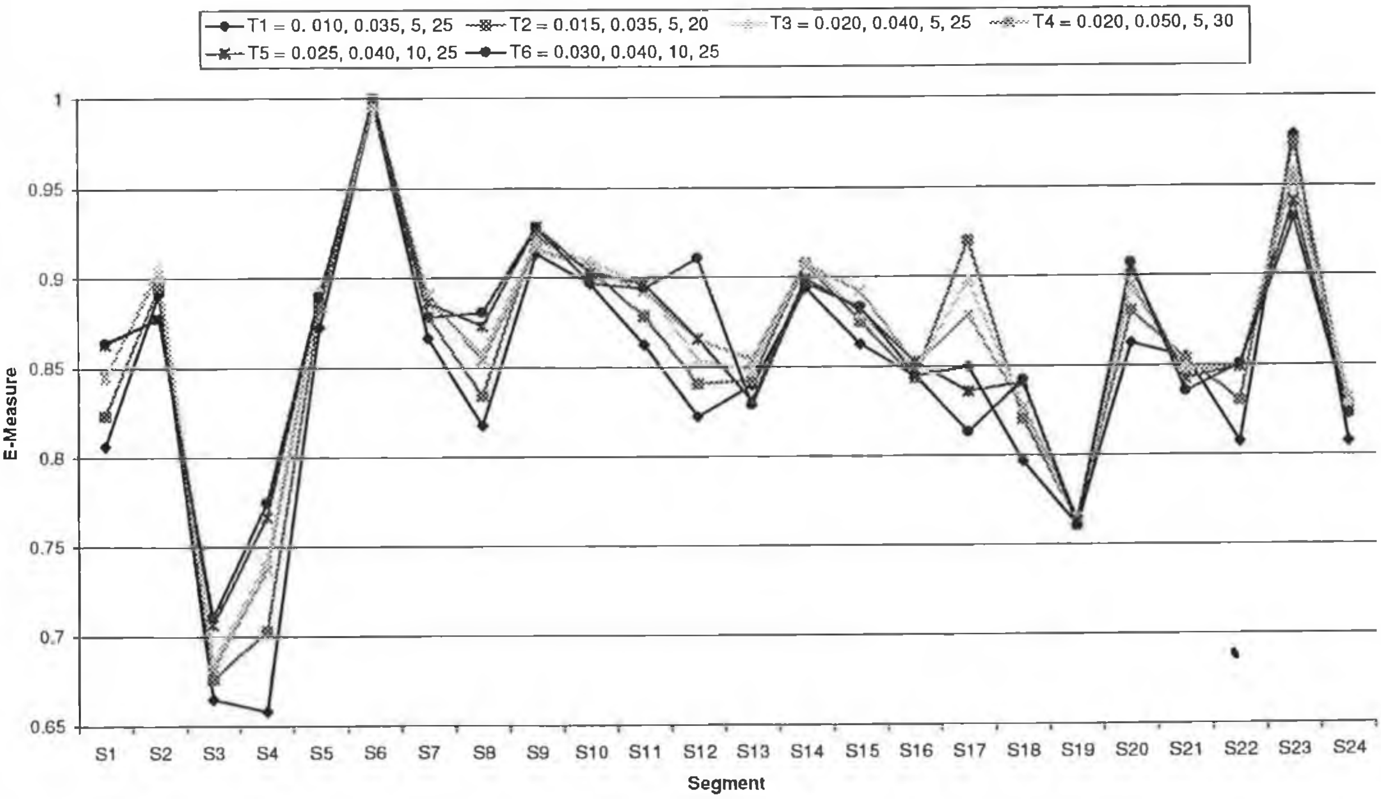


Figure 6-4: E-measure for 24 video segments using 6 combination thresholds

6.3.3 Do Varied Video Content Types Affect the Results Obtained from Different Fixed Thresholds?

We have seen the results obtained by a selection of shot boundary detection thresholds on the 24 segments of the video test set. However, these results tell us little about why a particular segment/threshold combination is producing a particular result. Our second set of experiments explored how effective the system was at segmenting specific content types. This would show how different content type/threshold settings interacted and affected the overall result.

This second experiment requires that we examine the video test set by video content type, rather than by segment, as each segment contains a mix of content types. We employed the same seven threshold settings as for section 6.3.2. Figures 6-5 and 6-6 show the recall and precision graphs for the eight video content types contained in the test set. Figure 6-7 shows the E-measure graph. The following general points can be noted:

1. The combination method generally strikes a balance between the high precision/low recall of the histogram algorithm and the low precision/high recall of the moments algorithm. This is the balancing effect that was discussed in section 6.3.2 above. More specific details are noted in the analysis of the specific content types below.

2. Poorest results were again achieved for the cookery and documentary content types. In the case of the cookery content type, recall was good, but precision remained poor, due to detection of false shot boundaries. The Documentary content type, in contrast, exhibited good precision but very poor recall, indicating that the shot boundaries present are too subtle for our current detection system.

3. At the current high levels of precision and recall (85%+), it is very difficult to make substantial gains in accuracy without an inordinate amount of effort. Thus, even though the combination system only results, on average, in gains of 2%-4% for precision and recall, we believe these gains to be highly significant. Also, we feel that to increase the accuracy of our colour-based system when applied to certain content types, and the shot boundaries they contain, would be difficult. This difficulty is a result of both inherent weaknesses in the underlying colour detection systems, and the basic methods of shot boundary threshold selection, when applied to a large and heterogeneous test suite. This problem will be explored more fully in chapter 7.

We can also comment on the different video content types. Note that when the results for the combination algorithm are compared to those of the histogram or moments algorithm, the threshold setting that results in optimum recall and precision scores is used for the latter two techniques. For brevity, the recall and precision scores of a certain algorithm may be expressed as 70%/80%, indicating a recall score of 70% and a precision score of 80%.

9. Commercials: At optimum threshold levels (Threshold 3), the combination algorithm achieved a recall of 84% and a precision of 77%. This compares favourably with the histogram result of 79%/78%, and the moments results of 82%/77%. Recall scores change little (±2%) as the threshold is raised or lowered, but lower threshold settings result in a reduction of precision to 68%, and so are not suitable.

10. Soaps: Again, threshold 2 proved to be optimal for this content type, resulting in recall and precision scores of 91% and 88% respectively. This is again, on average, superior to the histogram result of 82%/92% and the moments result of 76%/79%. The results for this content type change only slowly as the threshold is altered, indicating that a large number of shot boundaries are well defined and easy to detect. Both recall and precision remain in the high 80s and 90s throughout the range of thresholds.

11. News: This content type produced good results when analysed with the combination algorithm. High recall (90%+) and precision (82%+) scores were obtained across the entire range of thresholds. As such there is no one optimum threshold as different thresholds simply trade off recall and precision equally. However, taking threshold 4 as a sample result, we can see that the scores of 92% recall and 87% precision compare well against the histogram algorithm (87%/86%) and the moments algorithm (87%/89%)

12. Cookery: In contrast to some of the other content types, cookery produced quite varied precision results (43%-74%) as the threshold level changed. However the recall scores remained rather constant (83%-89%), indicating that the inability to detect the remaining shot boundaries may be due to inadequacies in the underlying colour-based detection algorithm rather than in the threshold selection. The optimum threshold settings are quite high, as they improve precision greatly without affecting recall too much. Threshold 7 results in a recall of 83% and a precision of 74%. Although these results leave much room for improvement, they compare favourably against those produced by the histogram algorithm (83%/71%) and by the moments algorithm (75%/63%).

13. Magazine/chat show: This content type produced very good results across the threshold spectrum. Recall was almost constant at 95%-96%, while precision varied only slightly more (89%-93%). As for the previous content type, the inability to raise recall above this level implies that the threshold selection is not the bottleneck in this case. Taking threshold 3 as the optimum, we can see that the recall and precision scores of 96% and 93% are superior to those produced by the histogram algorithm (94%/93%) and the moments algorithm (93%/87%).

14. Quiz: As expected from the results obtained in chapters 4 and 5, the quiz content type proved quite easy to segment accurately. Recall levels were again almost constant at 98%-99%, with precision ranging from 92%-98%. As with the news content type, there is no one optimum threshold as often we simply trade off recall and precision equally as we move along the range of thresholds. However, the higher threshold levels (from threshold 3 to threshold 7) produce better results than the lower ones, as precision starts to suffer as the threshold is lowered beyond a certain point. In the higher threshold band, there is absolutely no change in either recall or precision, irrespective of which threshold we choose both remain at a steady 98%. Achieving a recall or precision of 100% is quite possible with this content type, but would unbalance the overall result too much to be useful. To compare with the earlier techniques, the histogram method achieved a score of 97%/98%, while the moments methods resulted in scores of 98%/93%. Thus we can see that the combination method does not greatly improve upon the histogram method, but is significantly better than the moments method.

15. Comedy/Drama: While not as successful as for the last two content types, the combination algorithm performs quite well when applied to this style of video. Recall values ranged from 90% to 84%, while precision varied from 90% to 95%. Moving to threshold values outside this range leads to an increasingly uneven trade off between recall and precision, which is not suitable for our chosen application. Threshold 4 gives optimum recall and precision (but only by 0.05% above threshold 2) of 89% and 93% respectively. These results are again superior to both the histogram (89%/92%) and the moments (78%/93%) algorithms.

16. Documentary: As for the previous detection algorithms, the documentary content type proved to be the most difficult test for the combination method. Recall scores are very poor, ranging from 27% to 54%, while precision is acceptable at 77% to 80%. As for the cookery content type, a large part of these results must be attributable to the inadequacies of colour-based methods when applied to complex shot boundaries contained in black and white footage. Taking threshold 1 as the optimum, the results of 54% recall and 77% precision compare well with those produced by the histogram method (52%/57%), but are inferior to those produced by the moments algorithm (60%/75%). Interestingly, this is the only content type where the moments algorithm produces better results than both the histogram and the combination algorithms.

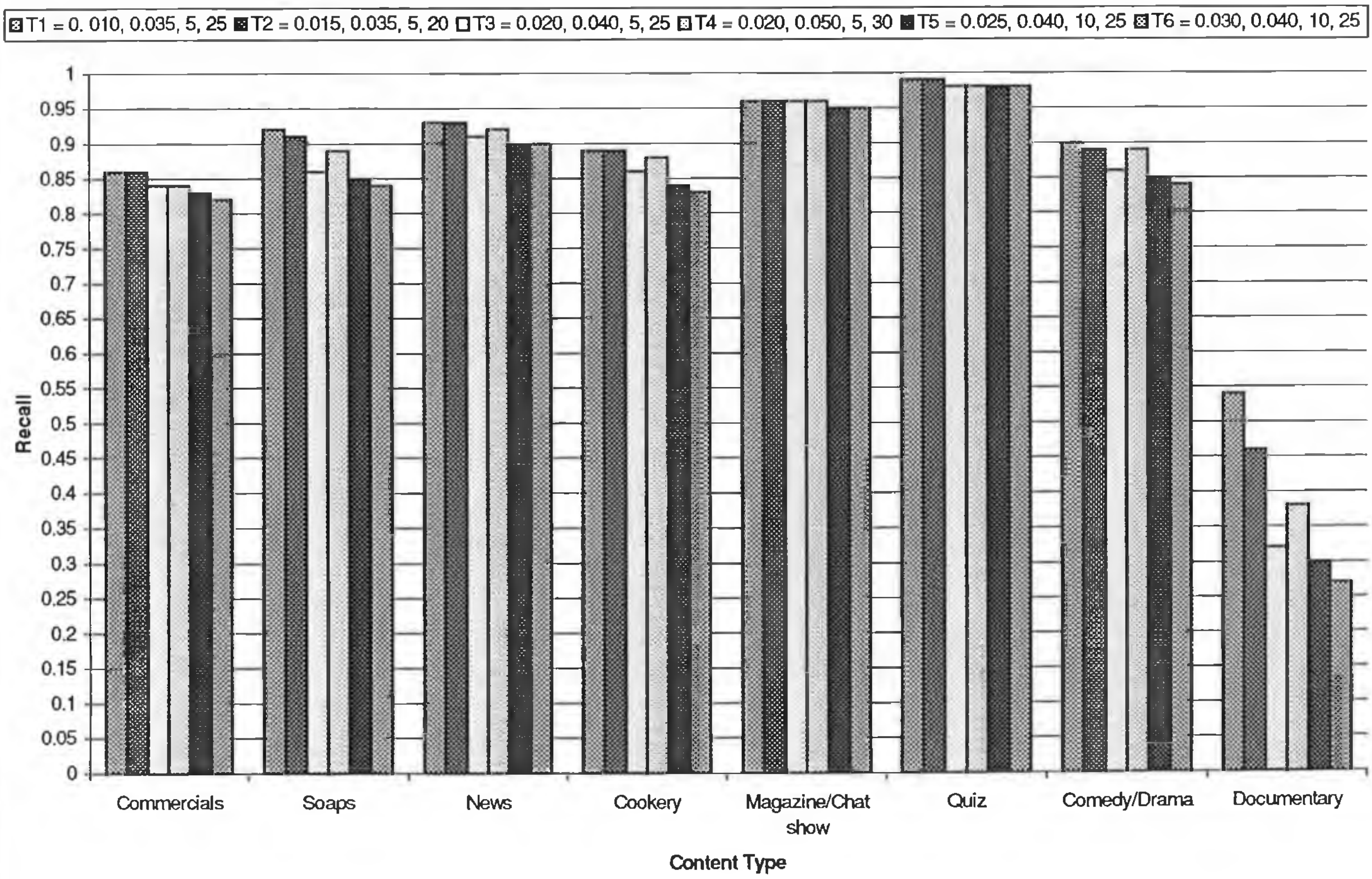


Figure 6-5: Recall for 8 video content types using 6 combination thresholds

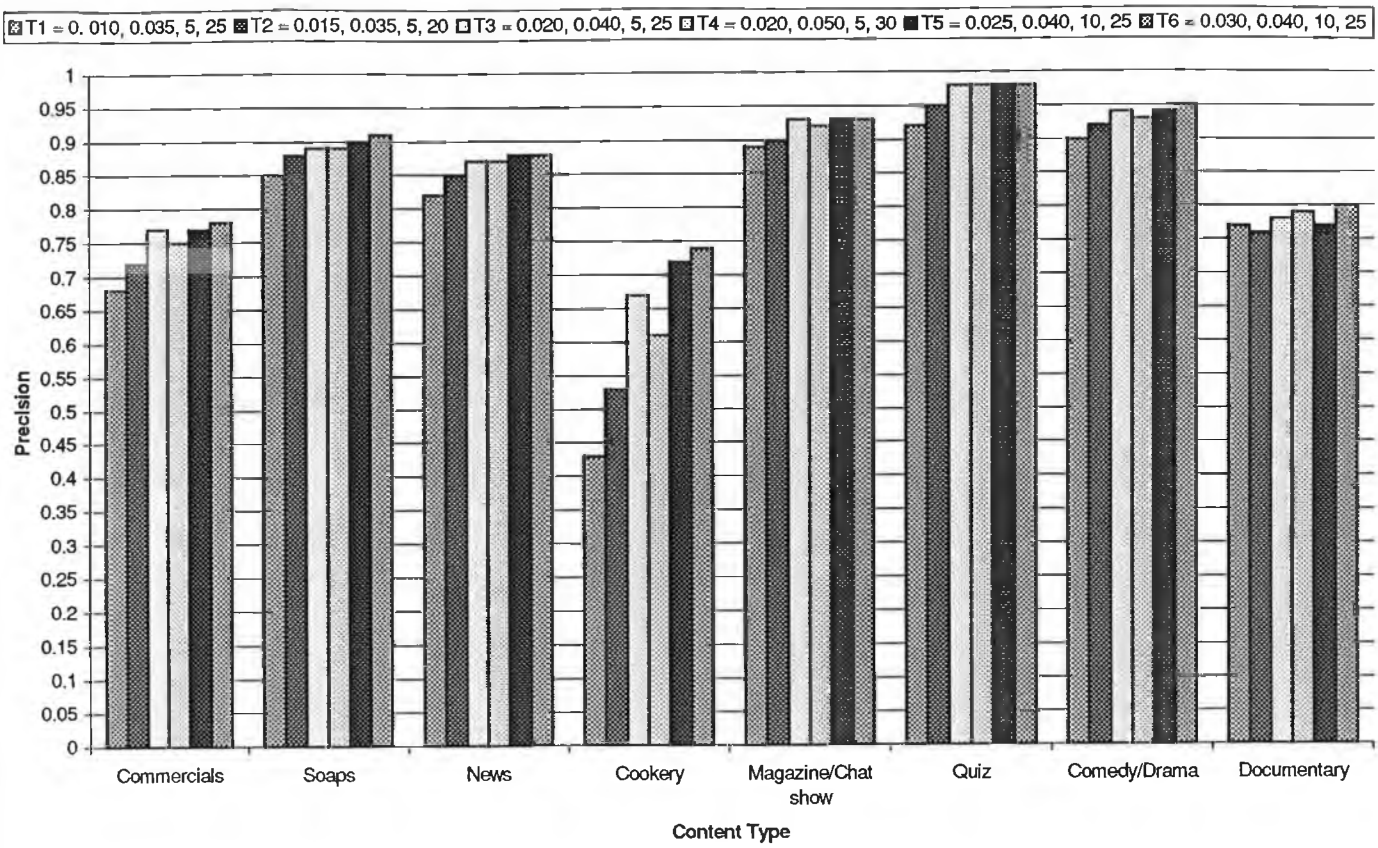


Figure 6-6:Precision for 8 video content types using 6 combination thresholds

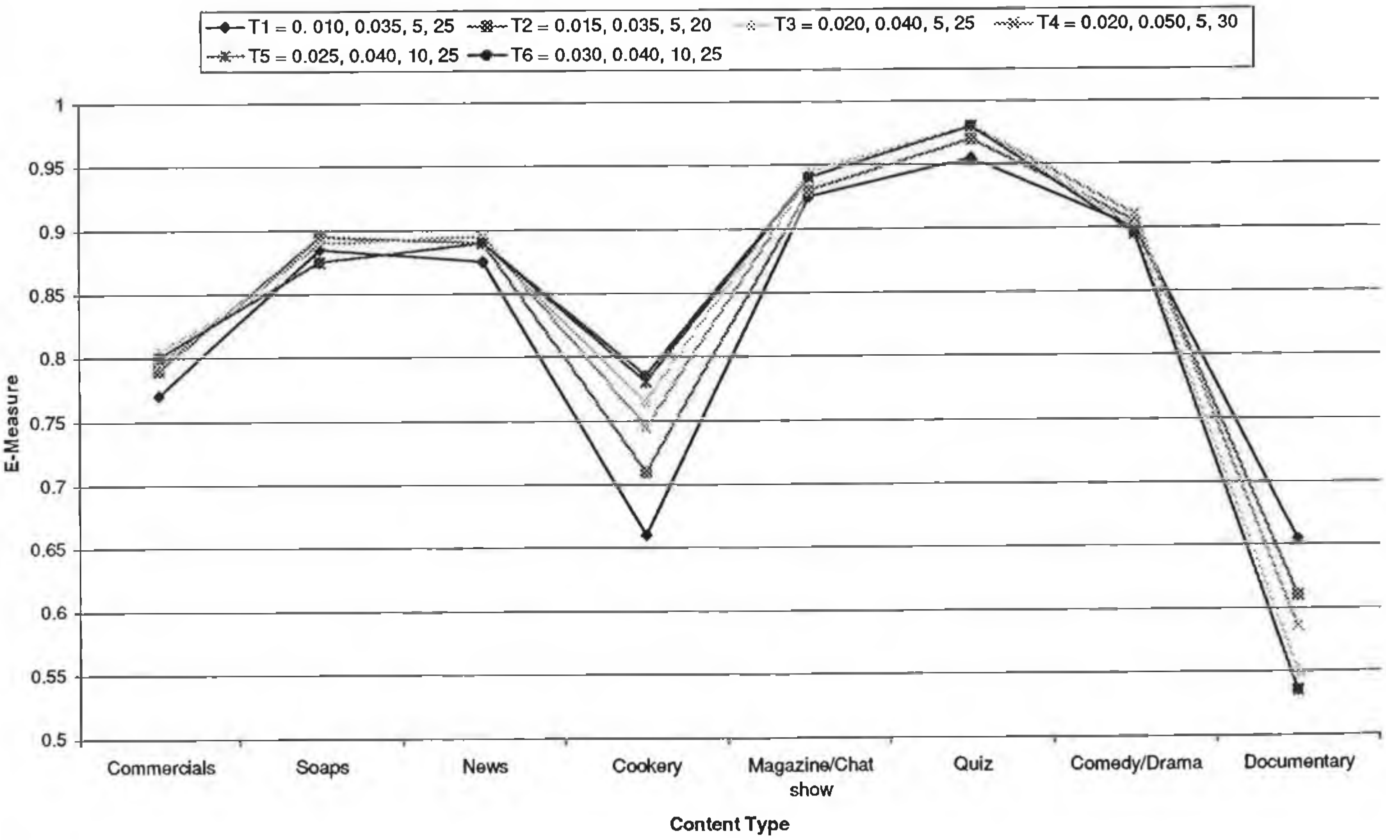


Figure 6-7:E-measure for 8 video content types using 6 combination thresholds

6.4 Summary

This chapter described the third shot boundary detection system developed during our research, based on a combination of the systems described in chapters 4 and 5. The method of evidence combination was defined, along with the assumptions upon which it is based.

In presenting our results, we once more employed precision, recall, and the E-measure as measures of segmentation accuracy. We presented the results obtained by evaluating this system upon the video test suite described in chapter three, along with analysis of the systems performance when applied to single, and multiple content types. We noted that the system performance was, in almost all cases, superior to that of either of the previous two algorithms. However, like the previous algorithms, we also noted that different content types often required different threshold settings to accurately segment.

We described the difficulty of raising a systems performance beyond a certain level without a great deal of effort for little return. This has often been expressed informally as the 80%/20% rule, indicating (for our purposes) that 20% of the time (or effort) is spent achieving 80% of the accuracy, while the remaining 80% of the time is spent trying to achieve the final 20% accuracy. We also noted that certain content types in our test suite are simply not suitable for segmentation by a colour-based algorithm, and would perhaps be better dealt with by using an alternate method of shot boundary detection.

The next chapter focuses on the problems inherent in attempting to accurately segment a large and heterogeneous video test suite using fixed threshold values. We introduce the concept of adaptive thresholding and describe a basic implementation for use with the three shot boundary detection algorithms described so far.

7.0 Adaptive thresholding

7.1 Introduction

During our research, we have identified several problems associated with the evaluation of video shot boundary detection techniques. One of the main problems identified is the common reliance on inadequate video test suites. Once a large and varied test suite is employed during experiments, weaknesses begin to appear in the traditional fixed difference threshold method of segmentation.

Drawing on the results obtained in chapters 4 to 6, this chapter describes a possible approach to modifying a fixed threshold system for use on a test suite containing multiple content types. We first focus on the challenge of segmenting such a test suite, then propose a method of adaptive thresholding. Finally, we describe a basic attempt to implement adaptive thresholding using the systems already developed during our research.

7.2 Difficulties with current segmentation systems

As described in the previous chapter, the combination algorithm achieves relatively high levels of segmentation accuracy. At these levels of precision and recall (85%+), it is very difficult to make substantial gains in accuracy without an inordinate amount of effort. This problem has been expressed in the previous chapter as the 80%/20% rule, indicating (for our purposes) that 20% of the time (or effort) is spent achieving 80% of the accuracy, while the remaining 80% of the time is spent trying to achieve the final 20% accuracy.

After much testing, we have noticed that our systems generally reach a certain performance ceiling, especially when applied to video segments comprising of multiple content types. The difficulty of improving our results beyond this approximate level may be due to a combination of the following facts:

- The inability of the actual detection systems to discriminate between a real shot boundary and a false one. If the underlying colour detection system (whether histogram or statistically based) is not accurate enough to distinguish certain classes of shot boundary, then a complementary system is needed to raise precision and recall scores above their present values. This system would need to be based on a non-colour based detection method to be effective. Such methods include edge detection, macroblock counting, and analysis of motion vectors. For example, after analysis of the raw results of the poorly performing content types, we found that a large number of the missed shot boundaries in the documentary type had a colour distance value less than certain of the false shot boundaries in the same content type. This means that separation of false boundaries from true ones is impossible and so our current systems are incapable of detecting this class of subtle shot boundaries. A particular culprit in this respect is the cookery content type, which contains many transitions that simply do not register as significant colour distance measures.

The solution to this particular problem, is of course, to utilise shot boundary detection systems based on techniques other than colour similarities. As mentioned above, many such techniques exist, and could be incorporated into a combination-style system similar to that described in chapter 6. However, such an undertaking is beyond the scope of our current research.

- Conflicts between the automatically generated results and the manually generated baseline log files. As the log files were being compiled, there was much discussion over what exactly constitutes a shot change, when a gradual transition can be said to begin and end, and other such issues. We found that our manual indexers often disagreed strongly in certain cases, particularly in video sequences containing complex computer-generated effects such as morphing, which can appear to totally alter a shot without an actual "real" shot boundary occurring. After much discussion, a general consensus was reached on how to classify these difficult elements. However, it is apparent that there is not a direct relationship between what we, as people, perceive a shot boundary to be, and what a colour-based algorithm will decide given the same raw information. Therefore, we think it is inevitable that a system based on such low-level features as colour, shape or texture, will always encounter difficulties with video containing shot boundaries that may be classed as subjective. This effect applies particularly to the commercials content type, as complex computer effects are found most often in this content type.

This is a difficult problem to address, and has its roots in more complex issues of how humans perceive and interpret visual information. Given a complex medium such as video (or even still images), the decision of each individual person as to the structural, semantical and conceptual meaning of even a single piece of media will differ. To attempt to model this complex and highly subjective interpretation would prove incredibly difficult. This is especially true when we consider that most computer-based systems operate on low-level visual features, such as colour and texture, while people generally perceive visual data at a much higher, conceptual level. Attempts to bridge the gap between these two views require more advanced techniques than are currently available, and again, this problem is beyond the scope of our research.

- The inability of a fixed detection threshold to reliably detect shot boundaries over a video suite containing heterogeneous content types. Even if the underlying detection system is capable of assigning larger difference values to real shot boundaries, it is apparent that the magnitude of these difference values will vary both within, and especially between content types. The same can be said of the magnitude of any possible false boundaries caused by camera or object motion. A single fixed threshold cannot be expected to perform accurately beyond a certain point when faced with such a varied test suite. Increasing the precision and recall scores then, would involve the modification of the system to allow the threshold to change depending on the type of video currently being processed. We refer to this process of modifying thresholds based on the characteristics of the video itself, as *adaptive thresholding*.

Of the three performance-limiting factors discussed above, adaptive thresholding is the only one that has the potential to increase the performance of our colour-based systems without the introduction of new techniques not included in our research. This chapter, therefore, focuses on the topic of adaptive thresholding when applied to the combination system described in chapter 6. We first explain the difficulties of utilising static thresholds when segmenting mixed content-type video, and then introduce a basic method of adaptive thresholding to attempt to improve our previous results.

7.3 Shot boundary detection for fixed-threshold systems

7.3.1 Introduction

For any colour-based similarity measure, High distance values can indicate one of two things. Firstly, it can (and should) signal that a real shot boundary has occurred. Secondly, it could be the result of '*noise*' in the video sequence, which may be caused by fast camera motion, a rapid change in lighting conditions, computer-generated effects, or *anything that causes a perceptual change in the video sequence without being an actual shot boundary*.

As discussed in chapter 4 (section 4.2.4), different content types often require different thresholds to be segmented accurately. We shall use a small example to illustrate this fact, based upon results obtained from the histogram shot boundary detection algorithm.

Figure 7-1 shows the results obtained from a short 2000-frame segment (1 min, 20 sec) of video, taken from an episode of the soap "Home and Away". The cosine values are plotted on the Y-axis. The peaks indicate high colour difference values and therefore denote shot boundaries. In this particular segment all the shot boundaries are cuts, i.e. no gradual transitions occur. As can be seen, no shot boundaries occur until around frame 550. The small peaks and bumps represent the 'noise' mentioned above.

In this particular sequence it can be seen that the noise levels are quite low. This makes it easy to detect a real shot boundary using a fixed threshold, shown by the horizontal line at cosine value 0.05. The transitions themselves are also very distinct. Thus, these results represent the ideal conditions for correctly identifying shot cuts using colour-based detection methods.

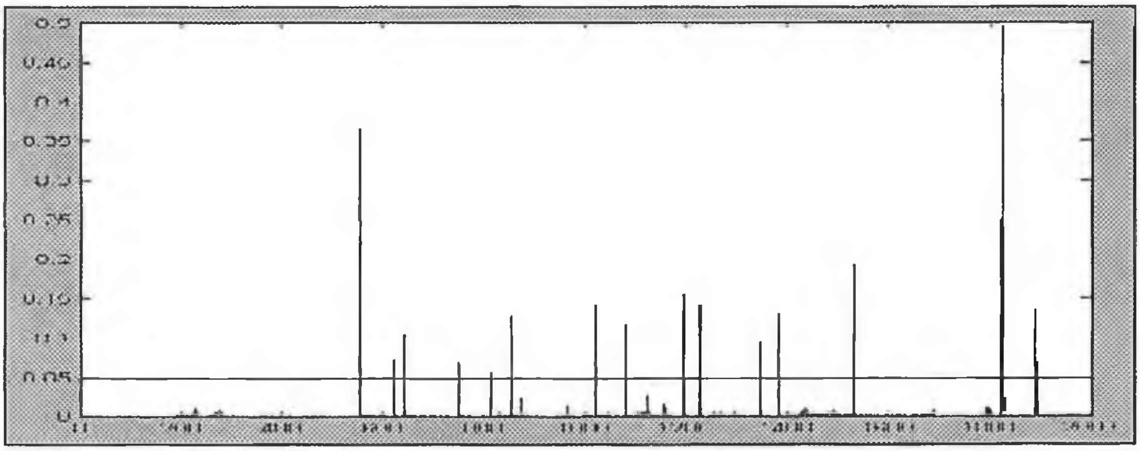


Figure 7-1. Cosine similarity results for 2000 frames of video from "Home and Away".

Unfortunately, these ideal conditions rarely exist in real-world television broadcasts, which is our target application environment. Modern television productions make extensive use of effects, including:

- Fades, dissolves and other gradual transitions.
- Computer-generated effects (e.g. morphing of one shot into another, especially in adverts).

- Split-screen techniques (e.g. ticker-tape, interviews, etc. where 2 or more "screens" appear on-screen).
- Global camera motion (e.g. zooming and panning shots which are used in almost all productions).

All of these techniques introduce noise into the video sequence, which may be either falsely identified as a shot boundary, or serve to mask the presence of real shot boundaries.

An example of the former case is a split-screen interview, as are common on TV news programs. In such cases the anchorperson remains constant in one window, with the second window switching between different reporters, and shots of the news event. The changes in the second window may indicate that a transition has occurred where in reality it is all one single logical video shot.

An example of the other effect of noise, where effects mask a shot cut, is the use of slow dissolves or morphs between scenes. In this case the change may be so gradual that the difference between consecutive frames is too low to detect.

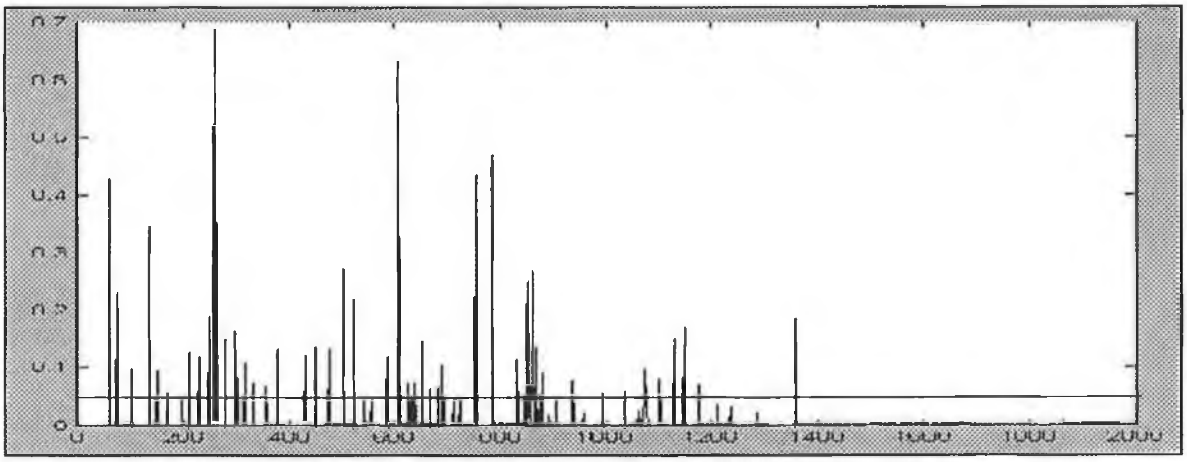


Figure 7-2. Cosine similarity results for a noisy segment of video.

Figure 7-2 is another 2000-frame sample. This piece of video is the end of a commercial break, returning to a program at around frame 1400. As can be seen, commercial breaks are usually noisy and hectic sequences. This is because, in comparison to programs, commercials typically have a huge number of cuts in a short space of time. Commercials also frequently include much more advanced visual effects than programs, frequently using computer generated techniques to distort, transform, and merge images. These facts make commercials some of the most difficult types of video to segment accurately. In contrast to figure 7-1, the same threshold (cosine value of 0.05) results in a large number of false positives.

7.3.2 Thresholds

To decide whether a shot boundary has occurred, it is necessary to set a threshold, or thresholds for the similarity between adjacent frames. Cosine similarity values above this threshold are logged as real shot boundaries, while values below this threshold are ignored.
To accurately segment broadcast video, it is necessary to balance the following two - apparently conflicting - points:

- The need to prevent detection of false shot boundaries, by setting a sufficiently high threshold level so as to insulate the detector from noise.
- The need to detect subtle shot transitions such as dissolves, by making the detector sensitive enough to recognise gradual change.

7.3.3 Analysis of optimum thresholds for content types

Table 7-1 shows the optimum threshold for the eight content types contained in the video test suite, for the combination system described in chapter 6.

Content type	Combination threshold
Commercials	Threshold 4
Soaps	Threshold 3
News	Threshold 3
Cookery	Threshold 6
Magazine/Chat show	Threshold 3
Quiz	Threshold 3 – Threshold 6
Comedy/Drama	Threshold 4
Documentary	Threshold 1

Table 7-1. Optimum combination thresholds for the 8 video content types.

It can be seen from the table that thresholds 3 and 4 are generally optimal. However, there are two exceptions to this rule

- The **cookery** content type performs best at a high threshold setting, as this increases precision without significantly lowering recall. Therefore we can say that the cookery content type contains elements, such as camera effects, object motion, and rapid illumination changes, that appear to indicate shot boundaries to the detection algorithm. As defined in section 7.3.1, we refer to such content types as *noisy*, that is, they cause a notable perceptual change in the video sequence without being an actual shot boundary.

  If we have a situation where the magnitude of the difference measure is greater for real shot boundaries than for changes caused by noise, then *raising* the shot boundary detection threshold to an optimum level allows us to detect the real shot boundaries while ignoring other perceptual changes. However, due to weaknesses in the underlying colour-based detection algorithm, many of the false shot boundaries in the cookery content type have greater distance measures than some of the more subtle real shot boundaries. In this case, raising the threshold level allows us to eliminate at least a number of false shot boundaries, and so increase precision, even if it does not significantly improve recall.

- The **documentary** content type performs best at extremely low threshold settings, not only for the combination algorithm, but also when analysed using the histogram and moments algorithms. In fact, as noted in chapter 5, the moments algorithm performs best on this content type. This is due to it's sensitivity to small changes in colour, which although a handicap when segmenting certain other content types, proves to be a bonus when dealing with this specific content type. The characteristics of this content type are a large number of distance measures of very low magnitude, so low in fact, that those thresholds that are optimal for other content types disregard them. Thus we find that, in the results presented for the documentary content type in chapter 6, lowering the threshold greatly improves recall (from 27% to 60%+) while lowering precision by only 2%-3%.

  This content type, therefore, may be considered as the opposite of the cookery content type, described above. While the cookery content type contains a large amount of noise in the video stream, which must be filtered out by *raising* the threshold, the documentary content type contains little noise, and real shot boundaries must be detected by *lowering* the threshold levels.

From studying these results, in conjunction with the example presented in section 7.3.1, we have developed a basic method of adaptive thresholding, based upon the level of noise encountered in a video segment. A description of this system is given below.

7.4 A method for adaptive thresholding

7.4.1 Introduction

Given the results presented in this chapter, and our general observations of the various shot boundary detection systems' performance on different content types, we now propose a basic method of adaptive thresholding. Attempting to design a system that automatically adapts to various conditions can be extremely challenging. Such adaptive systems can be of use in applications varying from region segmentation of still images (perhaps as a precursor to object identification), to control systems for real time embedded systems.

Given the limits of our research, we did not have the opportunity to pursue advanced adaptive methods, such as might be provided using neural nets or complex statistical models. However, as we feel that adaptive thresholding has the potential to be an important element in the accurate segmentation of digital broadcast video, we were interested to see if even a basic system could provide improvements over fixed threshold methods.

Even if such a system does not significantly improve the recall and precision results over a similar fixed threshold system using optimum threshold values, it is still a better choice for segmenting a heterogeneous video test suite. The reason is of course, that for a real-world automatic segmentation system, it would be impossible to manually select a fixed threshold for a particular content type. Therefore, if the algorithm itself can even approximate the optimum threshold selection a human would have chosen, it would be a major improvement over our current systems.

7.4.2 The relationship between noise and threshold levels

We have discussed, in section 7.3.3, how the levels of noise in a video sequence may give us indications of what type of threshold to use. By studying the cookery and documentary content type, we have illustrated how different content types may vary between containing subtle shot boundaries and high levels of noise caused by effects other than shot boundaries. Following from the definition given in section 7.3.1, we again define noise as *anything that causes a perceptual change in the video sequence without being an actual shot boundary.* Noise may be caused by fast camera motion, a rapid change in lighting conditions, or computer-generated effects.

Following on from this informal definition, we now define what we believe to be a general relationship between the levels of noise in a video sequence, and the corresponding threshold best suited to detect shot boundaries in that video sequence. Note that this definition only holds where the underlying difference algorithm is capable of assigning a greater difference measure to a real shot boundary than to a false one. As noted in section 7.2, if this is not the case, then the effectiveness of any adaptive thresholding techniques will be limited to those cases where the algorithm was able to correctly differentiate between real and false shot boundaries.

- Given a video sequence with a *high* level of noise, the shot boundary detection threshold should be *raised*, to prevent detection of false shot boundaries.

- Given a video sequence with a *low* level of noise, the shot boundary detection threshold should be *lowered*, in an attempt to detect subtle shot boundaries.

7.4.3 Using variance as an indicator of noise in a video segment

We can measure the amount of noise at a particular point in a video segment by focusing on the degree of variance at that point. If the variation about the mean distance measure for a particular segment of video is large, then the distance measures for different frames vary widely, and so that segment may be considered noisy. If the variation about the mean distance measure for a particular video segment is small, then individual frames have similar distance measures, and so that video segment can be considered to have a low level of noise.

Using a *sliding window* centred on the current frame pair, we can calculate a measure of variation for that pair. Figure 7-3 shows a graphical representation of the sliding window model.

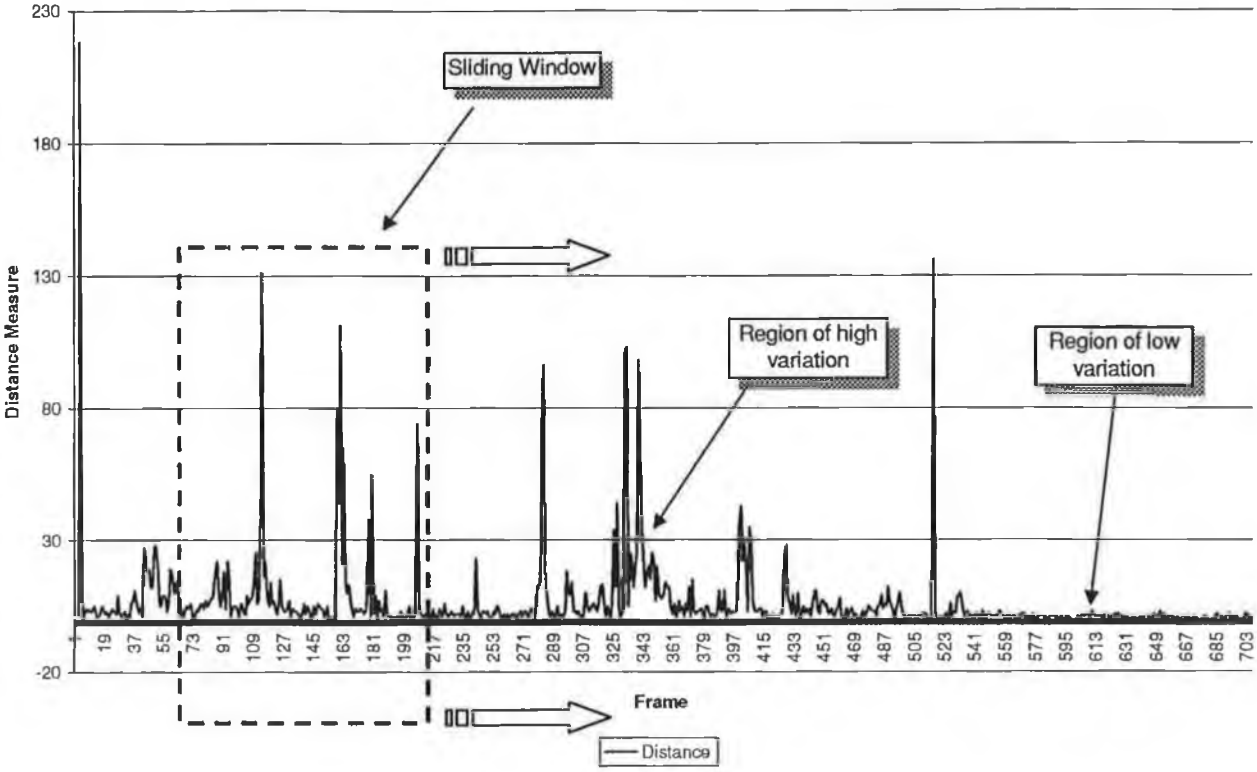


Figure 7-3. A sliding window used to calculate variance

As the mean will vary for each instance of the sliding window, we do not use the basic standard deviation measure. Instead, we calculate the *coefficient of variation V%*. This is simply the standard deviation normalised by the mean, the formula is shown below.

$$V\% = \frac{\sigma}{X} \times 100$$

Based on the rules defined in section 7.4.2, then, we can express the following relationship between the coefficient of variation and the shot boundary detection threshold. Assuming that we start the adaptive algorithm with a generally successful threshold level, for example, threshold 3 or 4 from the combination algorithm:

- High coefficient of variation ⇒ noisy segment of video ⇒ raise the shot boundary detection threshold to avoid false positives.

- Low coefficient of variation ⇒ quiet segment of video ⇒ lower the shot boundary detection threshold to detect subtle shot boundaries.

Of course, even using this simplistic method of approximating the amount of noise in a video segment, there are still several questions that must be answered:

1. What size sliding window should we use? A large window size may yield a more accurate result, but at the expense of increased time complexity. A smaller window may reduce look-ahead, and therefore accuracy, but will be more efficient.

2. How should we weight the frame values around the current pair? We may wish to employ some form of decay function to give higher priority to the values of frames closer to the centre of the sliding window.

3. Most obviously, what should be the change in the shot boundary detection threshold for a given change in the coefficient of variation? We may wish to implement a gradually changing threshold, equated somehow to the change in variation. Equally, we may decide to use a stepped system, where the threshold is set to one of a certain number of predefined levels, depending on the level of variation in the video segment.

We feel that the answers to these, and other questions, can only be found through experimentation, and "getting a feel" for the useful ranges of both thresholds and variance. Obviously, the experiments performed in chapters 4 to 6 have enabled us to choose useful threshold levels, but the relationship between the coefficient of variation and these threshold levels is still unknown.

7.5 Results

7.5.1 Aims and methods

This section presents results for the four-threshold based combination algorithm, as presented in chapter 6, modified to include the variance-based system of adaptive thresholding described in section 7.4 above. Our aims are simply to measure the effectiveness of a simple adaptive thresholding method, built on the combination algorithm, when applied to the video test suite, as compared to the version of the same algorithm employing fixed thresholds.

Our results for this chapter will be presented more informally than before, as this is basically exploratory work, and is presented more as a possible direction for future study, than as a complete system. As such, many of our implementation decisions are rather arbitrary, while others are decided by the small-scale experiments described below. Also, for simplicity, we simply present our results using the E-measure, employed as before as a weighted average of precision and recall. Refer to the results sections of chapters 4 and 5 for the definition of the E-measure metric.

Of the design decisions taken, the most important one was to employ predefined threshold levels depending on the current coefficient of variation levels. This approach was chosen because of the difficulty encountered while attempting to establish a gradual rate-of-change relationship between the coefficient of variation and the current threshold level. It is our opinion that using these predefined threshold settings allows a degree of control to the system designer, while still enabling the systems to adapt to different video types. Obviously, a more comprehensive study may determine that a gradual relationship between the two variables results in better system performance.

Given that we have decided on a stepped system of predefined threshold settings, the major questions that remain are what value these predefined thresholds should be, and what sliding window size to employ. These questions, which mirror those posed in section 7.4.3 above, are addressed in the following series of experiments.

7.5.2 Choice of sliding window size

One of the first questions to be considered is the size of the sliding window to be employed. For this first consideration we employed predefined thresholds shown in table 7-2, using sliding window sizes of 11, 41, and 81 frames to calculate the coefficient of variation. As explained in chapter 6, the combination algorithm uses an upper and a lower threshold level from each of the two methods employed. Therefore we define:

- Histogram_l: the lower threshold for the histogram system.
- Histogram_u: the upper threshold for the histogram system.
- Moments_l: the lower threshold for the moments system.
- Moments_u: the upper threshold for the moments system.

Coefficient of variation	Histogram$_l$	Histogram$_u$	Moments$_l$	Moments$_u$
0 to 100	0.010	0.025	3	10
101 to 200	0.015	0.030	4	15
201 to 300	0.020	0.035	5	25
301 to 350	0.025	0.040	7	30
350+	0.030	0.045	10	30

Table 7-2. Predefined threshold values used in the window size experiment.

Figure 7-4 shows the E-measure results for the predefined thresholds shown in table 7-2, when used with sliding window sizes of 11, 41, and 81 frames. Figure 7-5 shows the same information but for the eight video content types. As can be seen from the graphs, there is little difference between the different window sizes employed. As larger window sizes lead to slower execution times, we are encouraged to employ small window sizes by these results.

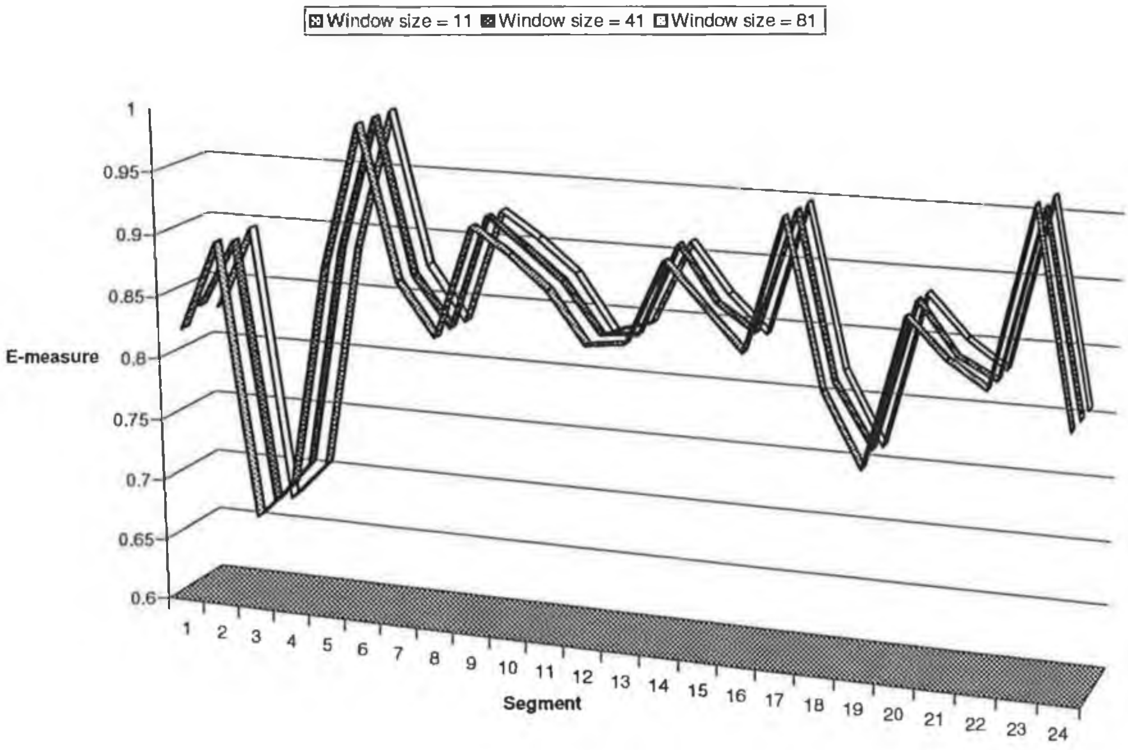


Figure 7-4:E-measure for 24 video segments using 3 sliding window sizes

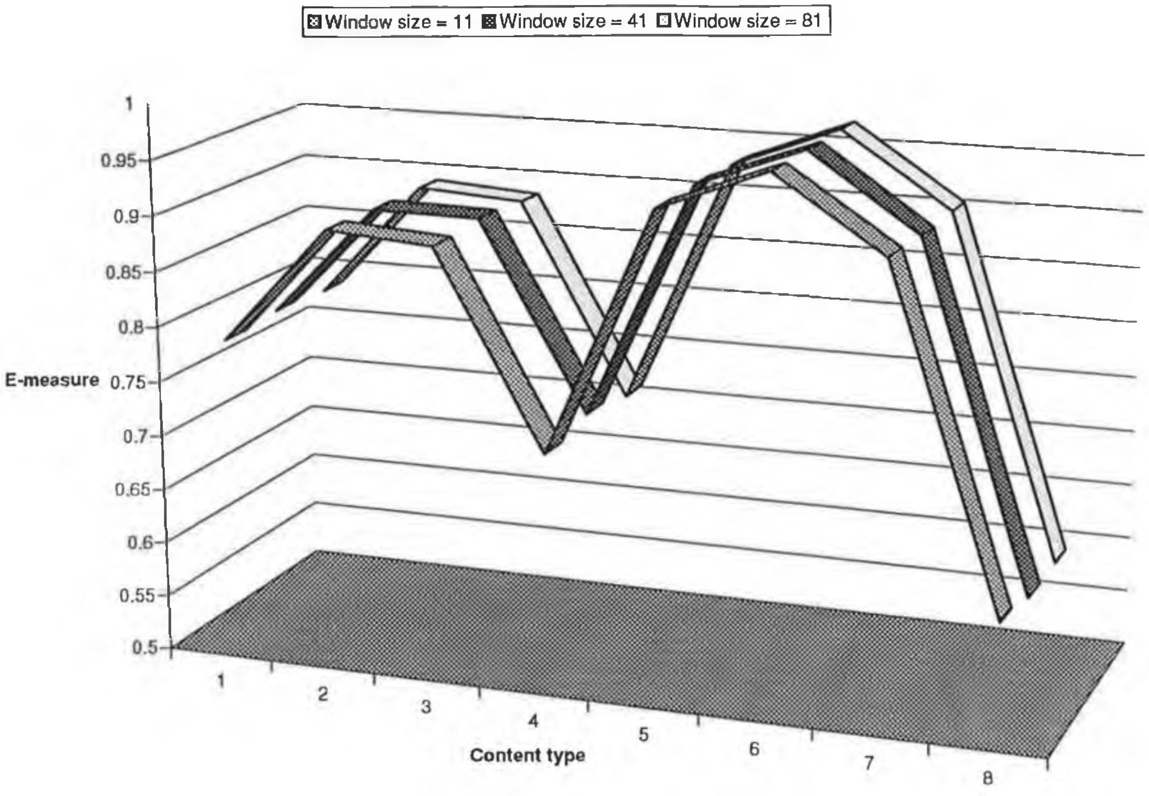


Figure 7-5:E-measure for 8 video content types using 3 sliding window sizes

7.5.3 Choice of predefined thresholds

We have established that a sliding window size of eleven frames is sufficient for our needs. The next, and obviously the most important design decision, is to define the relationship between the coefficient of variation and the shot boundary detection threshold. Table 7-2 (above) shows a starting relationship. However, the only way to determine the effectiveness of any threshold relationship is to evaluate it against the results of the original fixed threshold algorithm, and other adaptive threshold settings.

In our simplified system, finding the optimum relationship between variation and threshold levels is difficult, as in some sense we are "shooting in the dark". By this we mean that we can never quite be sure that we have reached an optimum value, or whether by altering some value or another we could make further improvements. This is similar to the problem of becoming trapped in local minima as experienced by certain types of neural nets. Whereas for neural nets the problem can be addressed by gradually lowering an originally high energy state until the global minima is found (a techniques used in Boltzmann machines, known as simulated annealing), in our case the only option available is repeated experimentation.

We used the combination system described in chapter 6 as a benchmark. Since threshold 3 (see section 7.3.3) was the optimum threshold overall, we chose to measure the results of the adaptive system against the results produced by the combination system using this setting.

Using perl scripts to control execution and collate results, we ran 40 experiments using a range of different variance/threshold settings.

- 29 of these experimental settings were dismissed instantly as they resulted in poor E-measure results as compared to the benchmark system.
- Of the remaining 11 systems, many are close to each other in performance. 6 of these systems give results comparable to the benchmark system. 2 systems result in very slight improvements in either precision or recall.
- The remaining 3 systems all produce noticeably superior results to the benchmark system, especially when applied to difficult video segments or content types.
- The best performing systems are those that use a threshold value close to that of the benchmark for general use, and only change from this setting once the variance raises or lowers beyond a certain point.

Table 7-3 shows details of the top performing adaptive threshold scheme. Figure 7-6 shows the E-measure results for this adaptive scheme and for the benchmark system, when applied to the 24 video segments. Figure 7-7 shows the same information for the 8 video content types.

Adaptive Scheme	Coefficient of variation	Histogram$_l$	Histogram$_u$	Moments$_l$	Moments$_u$
A1	0 to 50	0.010	0.025	4	15
	51 to 150	0.017	0.035	5	20
	151 to 250	0.020	0.040	5	25
	251 to 350	0.025	0.043	7	30
	351+	0.030	0.046	10	30

Table 7-3. Predefined threshold values used in the best performing adaptive system.

From results presented, we can note the following:

- The adaptive system improves the overall E-measure result of the benchmark system by 2%, from an average of 86% to 88% over the 24 video segments. Although this improvement does not seem large over the test suite as a whole, the improvements to individual content types and segments are more impressive, as discussed below.

- In figure 7-7, the content types which show the most improvement are those that require an unusually low or high threshold, specifically the cookery and documentary content types. The E-measure of the cookery content type is improved by almost 10%, from 76.5% to 85%. The E-measure of the documentary content type is improved by 12%, from 55% to 67%. The advantages of the adaptive system over the benchmark system are highlighted when dealing with such varied content types.

- It can be seen from figure 7-6 that segments containing these difficult content types, notably segments 3, 4 and 13, show a considerable improvement over the benchmark system. A clear example of this is segment 3, which has proved to be the most difficult video segment to accurately process. The adaptive system improves upon the results of the benchmark by 7%, from an E-measure result of 68% to 75%.

- With further work, it may be possible to further improve these results. However, it is worth remembering that even with an optimal adaptive thresholding system, we would soon reach another performance ceiling, caused by other factors, as discussed in section 7.2. Thus our efforts might better be directed towards improving performance through addressing one of these alternative methods.

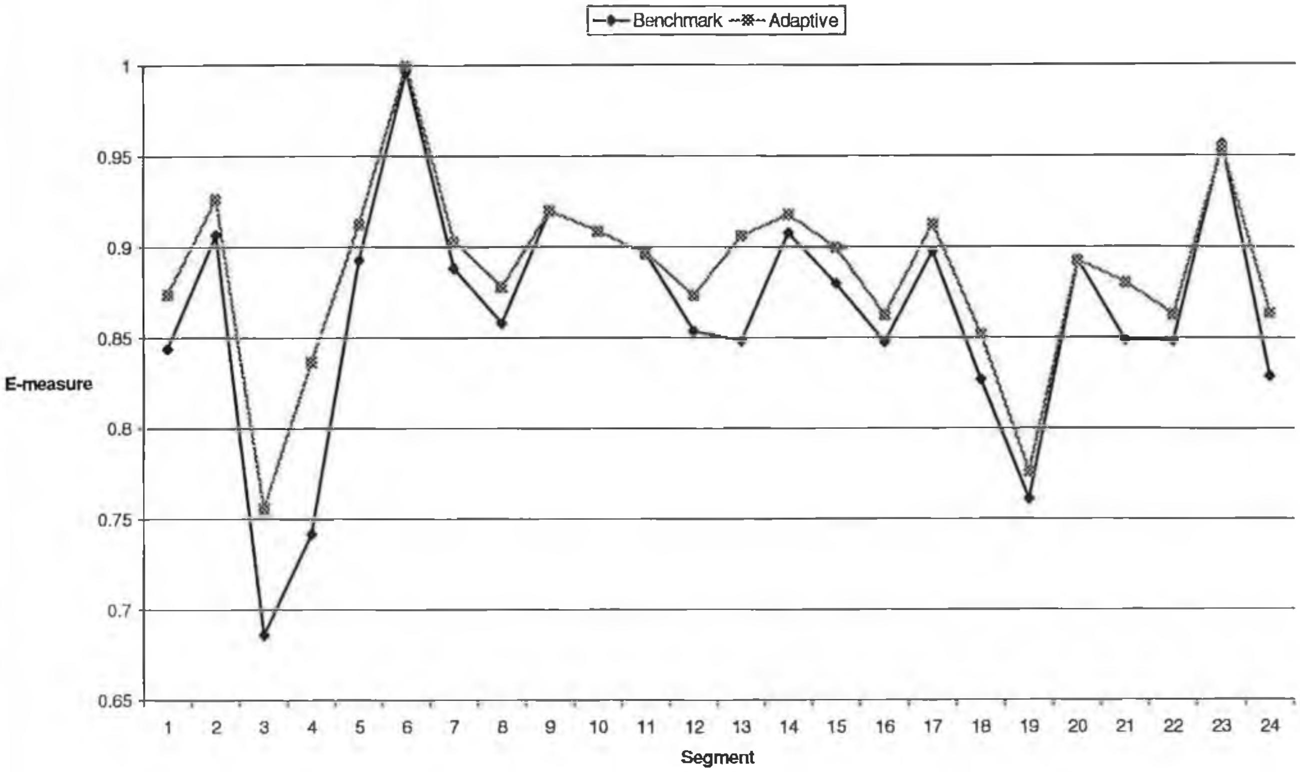


Figure 7-6:E-measure comparison between the adaptive and benchmark systems over 24 video segments.

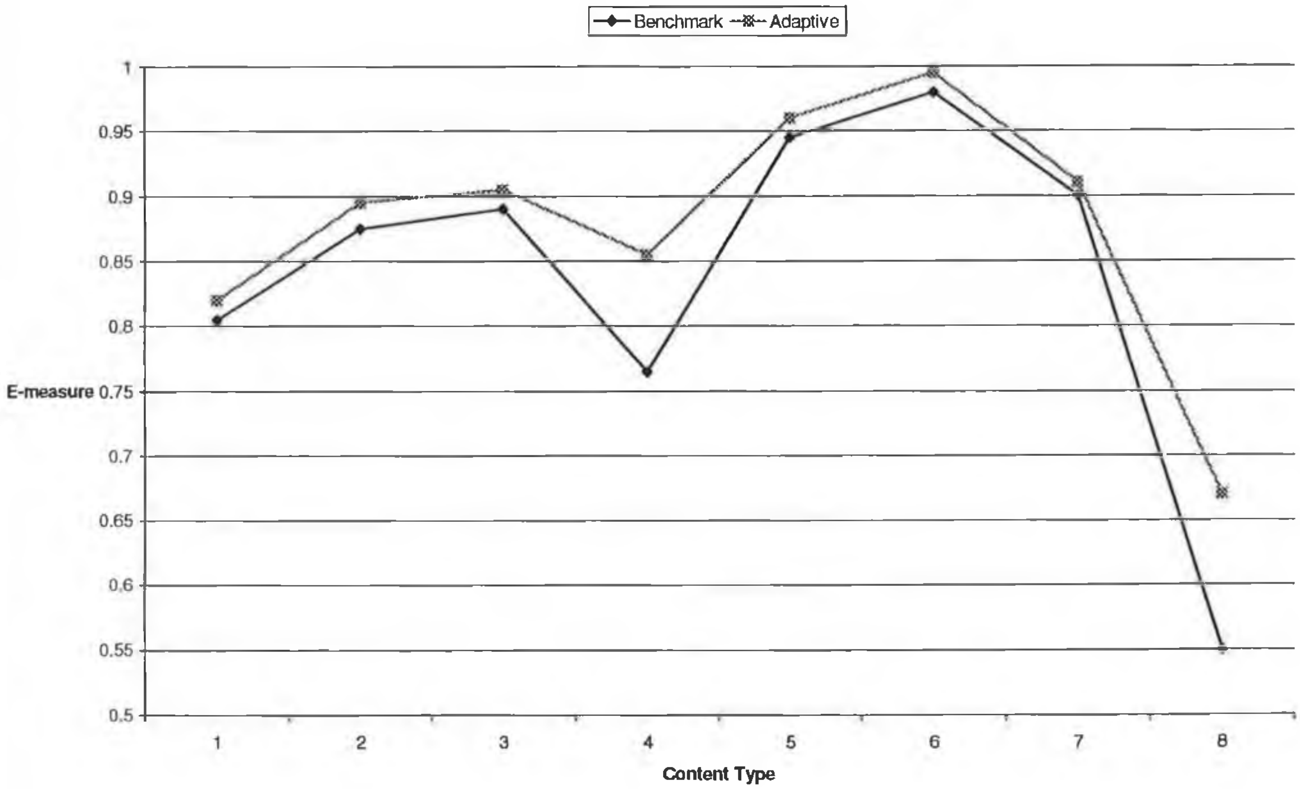


Figure 7-7:E-measure comparison between the adaptive and benchmark systems over 8 video content types.

7.6 Summary

This chapter described an experimental shot boundary detection system developed during our research, based on applying a method of adaptive thresholding to the combination system described in chapter 6. Adaptive thresholding is simply the altering of the system's shot boundary detection thresholds depending on the type of video currently being processed.

We first described the difficulty of raising a system's performance beyond a certain level without a great deal of effort for little return. We also noted that certain content types in our test suite are simply not suitable for segmentation by a colour-based algorithm, and would perhaps be better dealt with by using an alternate method of shot boundary detection.

We then highlighted the difficulty employing a fixed-threshold shot boundary detection system when attempting to segment a video test suite consisting of multiple content types. Based upon the results obtained in chapter 6, we detailed the optimum thresholds for the eight content types in the test suite, and found significant differences. We then examined the reasons for these differences, specifically when applied to the cookery and documentary content types.

Using these content types as examples, we defined a simple relationship between the level of noise in a particular segment video and suitable shot boundary detection thresholds. Following from this we described a basic method of adaptive thresholding, using the coefficient of variation to set suitable threshold levels.

In presenting brief results for this experimental system, we used the E-measure as a measure of segmentation accuracy. We found that systems employing even a basic form of adaptive thresholding can significantly improve upon the results obtained using systems with fixed thresholds.

The next chapter presents a summary of our research and details the conclusions reached during the course of this thesis.

8.0 Conclusions and future research

8.1 Introduction

In this thesis, we have investigated the application of colour-based shot boundary detection methods to a large and heterogeneous video test suite. We first presented related work in the field of video segmentation techniques, and described several real-world systems that have implemented these techniques. We noted that more rigorous evaluation of these systems is required, particularly with regard to the size and content of the test suite employed. Our objective was to develop a shot boundary detection technique based on colour similarities between frames, and then to comprehensively evaluate this system using a test suite representative of our target application – the automatic segmentation of broadcast digital video.

8.2 Summary of research

We firstly described the development of two systems based on different methods of measuring inter-frame similarity, namely colour histograms and colour moments. We highlighted the strengths and weaknesses of each method, and then developed a system that attempts to combine the best elements of each. In reporting the results for this combined system we found that it does indeed improve upon the results obtained by either of it's component systems.

During our experiments, we noted the difficulties of employing detection systems that rely on fixed thresholds, especially when attempting to segment varied video content types. We saw that different content types can require radically different shot boundary detection thresholds to be accurately segmented. As broadcast television consists of a multitude of heterogeneous content types, we identified a strong need for a more flexible system in this domain.

The concept of adaptive thresholding was introduced as a possible solution to this problem, adaptive thresholding being defined simply as the altering of a systems shot boundary detection thresholds depending on the type of video currently being processed. We defined a simple relationship between the level of *noise* in a particular segment video and suitable shot boundary detection thresholds for that segment. Following from this we described a basic method of adaptive thresholding, using the coefficient of variation to set suitable threshold levels. Results for this system demonstrated that even a basic method of adaptive thresholding has the potential to significantly improve shot boundary detection performance on a varied-content test suite.

8.3 Conclusions

In this section we comment on the conclusions reached, and lessons learnt, in the course of our research.

From analysis of the results generated by the four systems developed and evaluated, it is obvious that shot boundary detection systems can successfully utilise frame colour similarities to successfully segment broadcast digital video. However, this claim must be moderated by several observations:

1. The selection of a suitable test suite is crucial to the proper evaluation of any video segmentation system. The test suite employed during our research is both large and varied. However, it encapsulates but a fraction of the complexity that the broadcast television medium delivers. Thus, all that we can attempt to do is model this diverse medium as accurately as possible given our limited resources, and our results may not be applicable to video content types not included in our test suite.

2. In general, systems employing colour frame similarity measures can detect abrupt cuts and relatively simple shot boundaries with a high degree of accuracy. However, the broadcast television medium contains content types that prove much more challenging, both in terms of the complexity of actual shot boundaries present, and also the visual effects present. When utilising fixed threshold systems, namely the histogram, moments, and combination systems, we have seen the difficulty of striking a balance between successfully detecting subtle shot boundaries, while at the same time ignoring false indications caused by other visual effects. Of course, this is but a specific instance of the wider problem of balancing precision and recall in traditional Information Retrieval systems.

3. In attempting to accurately segment video into its constituent shots, we have seen that there is no optimum method that ensures success when applied to a sufficiently varied test suite. Rather, the best approach is to employ a combination of different approaches, in an attempt to compensate for the inadequacies of certain techniques with the strengths of

others. Although our combination system incorporated only colour-based techniques, it still managed to improve upon the results obtained by either of the single-techniques systems that preceded it. A system that combines techniques based on different features, such as colour and shape, therefore, could be expected to improve matters even further.

4. The reliance of many systems on fixed shot boundary detection thresholds is a serious weakness when applied to a varied test suite. Although manual selection of thresholds may be appropriate for certain applications, it is ultimately limiting when attempting to design systems for use in non-domain specific roles. We strongly feel that adaptive thresholds, together with multiple evidence combination (if possible from systems based on automatic extraction of different features) are necessary requirements given the complexity of the broadcast television medium.

The four systems developed gave us valuable insight into the practicalities of detecting shot boundaries. From starting with an understanding of the issues involved with the analysis of still images, we quickly gained an appreciation of the challenges inherent in the video medium. In particular, we soon realised the need for extremely efficient analysis of individual frames – techniques suitable for still images tend to result in unbearably long execution times when applied to eight hours of video data.

Following are some general conclusions and final comments on the specific systems developed.

8.3.1 The histogram-based system

We chose histograms as the basis of our first system because of their accuracy and ease of computation. As such, the results obtained reflect almost exactly the strengths and weaknesses of this method. General shot boundary detection was good, and the quantization of the original pixel values help to insulate the system from noise caused by visual events in the video segment. However, this quantization also made detection of gradual transitions difficult, and produced poor recall results for content types that contained many such transitions, notably the documentary type.

8.3.2 The moments based system

We chose moments as the basis for our second system as they have been reported as among the most successful statistical-based methods [1, 23]. We found that, in contrast to the first system, the moments-based system was more affected by non-shot boundary related visual effects in the video stream. However, this sensitivity also led to increased recall, as subtle shot boundaries were more readily detected. As such, the moments systems has different strengths and weaknesses than the histogram system, and so the appeal of combining them was obvious.

8.3.3 The combination system

Combining our first two systems produced results that were significantly superior to either of them individually, although the constraints of utilising fixed shot boundary detection thresholds limited its effectiveness. Typically, the system resulted in precision scores similar to the histogram method, coupled with the increased recall of the moments method. The combination system outperformed both individual systems over all eight video content types, indicating that it is much more robust with respect to both detecting subtle shot boundaries, and discarding misleading visual effects. One drawback of the combination algorithm is the time taken to process the video files. Whereas the histogram and moments algorithms execute in real time, the combination algorithm requires one and a half times real time to process a video segment.

8.3.4 The adaptive system

Considering the basic method of adaptive thresholding that we employed, the resulting improvements in performance indicate that the potential benefits of a more complex implementation could be significant. The adaptive system performed similarly to the combination system (using the optimum fixed threshold value), except when processing content types that exhibit significantly different characteristics to the norm. On these content types, of which we feel sure there are many in real-world broadcast television, the adaptive systems comprehensively outperformed it's more basic counterpart. Given that execution time is not significantly greater than for the basic combination system, and that the results are universally superior, we see no reason to return to a fixed threshold scheme.

Figure 8-1 shows an overall comparison of E-measure results between the four test systems based on the optimum results obtained from each.

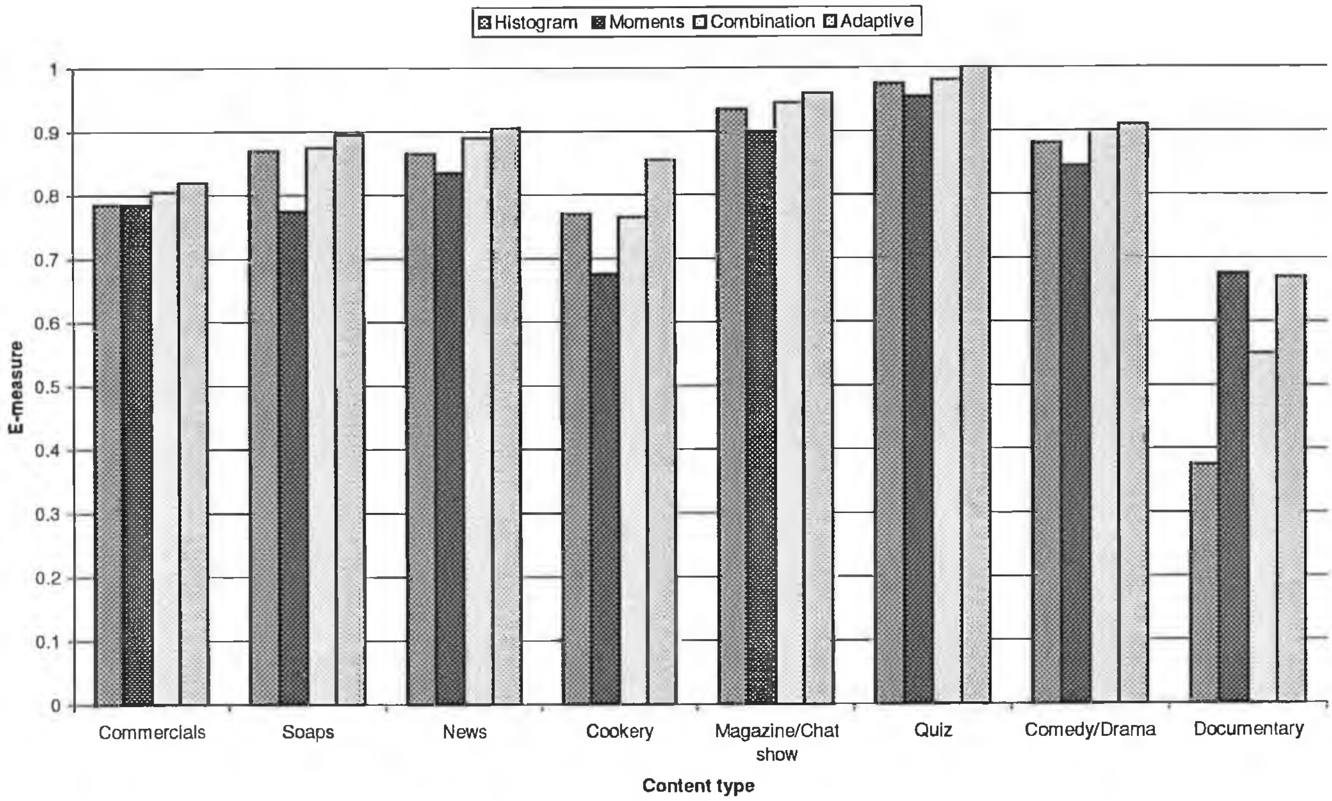


Figure 8-1. A comparison of the four test systems using optimal thresholds.

8.4 Future work

The area of digital video segmentation is so vast and diverse that there are countless areas for future work, both in improving the systems described in this thesis, and in implementing complementary systems based on features other than colour.

An obvious improvement to the current systems would be the implementation of a similarity measure across multiple frames, to aid in the detection of gradual transitions. Such methods, of which the running histograms technique [3] is an example, allow for greater accuracy in detecting transitions such as fades, dissolves, and wipes. However, although they therefore improve recall, as more shot boundaries are identified, they can also adversely affect precision, as differentiating between such subtle transitions and spurious noise caused by video effects can be difficult.

The topic of adaptive thresholding is one that we have focused on only briefly, though we believe it is a natural, and indeed necessary, progression from the current fixed threshold systems. The development of a comprehensive and robust method of automatically tailoring shot boundary detection thresholds to particular video types is a promising direction for future work. This method may be based upon neural net technology, such as Hopfield nets or Boltzmann machines, as these systems have the ability to automatically learn and respond in certain ways to input patterns. Equally, it may involve a statistical or mathematical representation utilising certain identifiable variables extracted from different video types. Although we employ the variance of the frame similarity values for this purpose, there are surely more comprehensive measures that can be exploited.

Execution speed is also an important feature of any system concerned with processing digital video. Although the current systems are reasonably efficient, even without such optimisations as might be made, there is room for considerable improvement. Recall from chapter one, that MPEG is a compression algorithm, where raw video is encoded in order to make more efficient use of bandwidth. Our current systems operate upon the decoded MPEG video sequence. This means that before the video can be analysed and segmented, it must be first decoded so as to access the pixel values of the original frames. This decoding is quite a computationally expensive process, and accounts for much of the total execution time of the current algorithms.

Referring to section1.3.4, we can see that the MPEG algorithm employs the Direct Cosine Transform (DCT) to reduce spatial redundancy. It is possible to retrieve average colour values for blocks of pixels by analysing the DC coefficient of the DCT, without having to fully decode the MPEG bit stream. Thus, with some modifications, our algorithms could operate on the encoded bit stream (analysing encoded pixel block values, rather than the original pixels), thus removing the need for the computationally expensive decoding process and greatly improving performance. We have utilised this method of colour analysis on still images stored in JPEG format as part of another research project, and found that execution time was reduced by several orders of magnitude, while discriminatory capacity was virtually unchanged. Although an MPEG bit stream is more complex than a single JPEG image, due to the method of temporal compression using inter-frame coding techniques, the same principles can be applied.

Moving away from our current systems, the next obvious step is the development of systems based on alternative methods of detecting shot boundaries, and the application of various combinations of these systems to discover how they interact when applied to varied video types. Obvious choices for alternative systems are those techniques listed in chapter two, including edge detection, shape recognition (although this is a difficult problem in still images, let alone video), and features extracted from the encoded bit stream, such as motion vectors and macroblock proportions. These last two methods, as well as an edge based one, have been developed by other researcher working on the same project that our research is part of. These methods will be evaluated using the video test suite described in chapter three, and comparisons and possible combinations of all of the resulting systems is planned.

Although our research has focused on methods of shot boundary detection, it is obvious that accurately detecting shots is only the first step in processing a video stream for presentation to a user. The ability to automatically group individual shots into scenes is a desirable feature of any digital video system. As noted in chapter one, people generally visualise video as a sequence of scenes not of shots, similar to a play on a stage, and so shots are really a phenomenon peculiar to only video. Thus, a system which presents a user with a video stream segmented into both shots and scenes is far more likely to satisfy his or her information need than a system reliant solely on shots. Grouping of shots into scenes is still very much an open research issue, but it is clear that is requires some semantic understanding of the contents of this video to be successful. Attempting to construct scenes using only primitive features such as colour and texture are unlikely to succeed outside a limited domain. To devise a reliable system for this purpose will likely require analysis of multiple sources of evidence, perhaps including visual cues, audio extracted from the soundtrack and analysed using speech recognisers, and temporal data such as the number and duration of individual shots.

Digital video is perhaps the fastest-growing medium in the world today, and certainly one of the most complex to process automatically. Our research has focused on methods to enable developers, authors, and end users to easily take advantage of its random-access nature, and therefore improve the way they use this exciting medium, both for business and leisure purposes.

9.0 References

1. G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, in Journal of Visual Communications and Image Representation, volume 7, pages 28-43, March 1996.

2. J. Boreczky and L.A. Rowe, Comparison of video shot boundary detection techniques, in *IS&T/SPIE proceedings: Storage and Retrieval for Images and Video Databases IV*, volume 2670, pages 170-179, February 1996.

3. X. U. Cabedo and S. K. Bhattacharjee, Shot detection tools in digital video, *in Proceedings Non-linear Model Based Image Analysis 1998,* Springer Verlag, pages 121-126, Glasgow, July 1998.

4. J. Canny, A computational approach to edge detection, in *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6), pages 679-698, 1986.

5. V. Castelli, L. D. Bergman, C.-S. Li, and J. R. Smith, Search and progressive information retrieval from distributed image/video databases: The SPIRE project.

6. S. F. Chang et al, VideoQ: An automated content based video search system using visual cues, in ACM Multimedia 1997.

7. S. F. Chang, J. R. Smith, M. Beigi and A. Benitez, Visual information retrieval from large distributed online repositories, in *Communications of the ACM*, volume 40, pages 63-71, December 1997.

8. M. G. Christel, Addressing the content of video in a digital library, in *Electronic Proceedings of the ACM Workshop on Effective Abstractions in Multimedia*, November 4, 1995.

9. M. G. Christel, D. B. Winkler, C. R. Taylor, Multimedia abstractions for a digital video library, in *Proceedings of the ACM Digital Libraries Conference*, Philadelphia, July 1997.

10. M. G. Christel, D. B. Winkler, C. R. Taylor, Improving access to a digital video library, in *Human-Computer Interaction: INTERACT97, the 6th International Conference on Human-Computer Interaction*, Sydney, July 14-18 1997.

11. M. G. Christel and D. Martin, Information Visualisation within a Digital Library, in *Journal of Intelligent Information Systems, information visualisation special issue*, June 1998.

12. Y. Gong, C. H. Chuan, and G. Xiaoyi, Image indexing and retrieval based on colour histograms, in *Multimedia Tools and Applications*, volume 2, pages 133-156, 1996.

13. A. Hanjalic, R. L. Lagendijk, J. Biemond, Achievements and Challenges in Visual Search of Video.

14. A. Hanjalic, R. L. Lagendijk, J. Biemond, A new key-frame Allocation Method for representing stored video streams.

15. J. Hunter and R. Iannella, The application of metadata standards to video indexing.

16. Kasturi and R. Jain, Dynamic Vision, in *Computer Vision: Principles*, pages 469-480, IEEE Computer Society Press, 1991.

17. W. Li, S. Gauch, J. Gauch, and K. M. Pua, VISON: A Digital Video Library, in Proceedings of ACM Digital Libraries 1996, March 1996, Bethesda, MD, pages 19-27.

18. W. Mahdi, L. Chen and D. Fontaine, Improving the spatial-temporal clue based segmentation by the use of rhythm, in Proceedings ECDL'98, Greece, Hearkleon, Crete, 19-23 September 1998.

19. J. Meng, Y. Juan, S.-F. Chang, Scene change detection in an MPEG compressed video sequence, in *IS&T/SPIE Symposium Proceedings*, volume 2419, February 1995.

20. A. Nagasaki and Y. Tanaka, Automatic video indexing and full-video search for object appearances, in *Visual Database Systems II*, pages 113-127, Elsevier, 1992.

21. N. V. Patel and I. K. Sethi, Shot video detection and characterisation for video databases, in *Pattern Recognition*, volume 30, pages 583-592, 1997.

22. M. Pilu, On using raw MPEG motion vectors to determine global camera motion, *Hewlett-Packard Digital Media Department Technical Report*, August 1997.

23. L. A. Rowe, J. S. Boreczky and C. A. Eads, Indexes for user access to large video databases, in Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases II, Pages 150-161, California, 1994.

24. C.J. Van Rijsbergen, *Information Retrieval*, London: Buttersworth, 1979.

25. T. Sikora, MPEG digital video-coding standards, in *IEEE Signal Processing Magazine*, pages 82-99, September 1997.

26. B. Simpson-Young and K. Yap, FRANK: Trialing a system for remote navigation of film archives, in *Proceedings of SPIE International Symposium on Voice, Video and Data Communications*, Boston, 18-22 November 1996.

27. J. R. Smith and S. F. Chang, Visually searching the web for content, in *IEEE Multimedia*, Pages 12-20, July 1997.

28. H. D. Wactlar, T. Kanade, M. A. Smith and S. M. Stevens, Intelligent access to digital video: Informedia Project, in *Computer theme issue on the US Library Initiative*, 29(5), pages 46-52, May 1996.

29. G. K. Wallace, The JPEG still picture compression standard, *in Communications of the ACM*, April 1991.

30. R. Zabih, J. Miller, and K. Mai, A feature-based algorithm for detecting and classifying scene breaks, in *Proceedings ACM Multimedia 95*, pages 189-200, November 1993.

31. H. J. Zhang, A Kankanhalli, and S. W. Smoliar, Automatic partitioning of Full Motion Video, in ACM/Springer Multimedia Systems 1(1), pages 10-28, 1993.

32. H. J. Zhang, C. Y. Low, S.W. Smoliar and J. H. Wu, Video parsing, retrieval and browsing: an integrated and content-based solution, in *Proceedings of ACM Multimedia*, November 1995.