Imperial College London
Department of Computing

Video Retrieval and Summarisation

Marcus J Pickering

July 2004

Submitted in partial fulfilment of the requirements of the PhD degree in Computing Science of the University of London.

Abstract

We present novel approaches to video shot and story boundary detection, and content-based retrieval using key frames. The techniques are deployed in a powerful system for retrieval and summarisation of broadcast news video

Shot boundary detection is an important technique for breaking video material down into units suitable for retrieval, and is fundamental to a video retrieval system. We present a novel algorithm that has been proven to perform highly accurately across a range of different types of video material. We extend this work for story boundary detection in broadcast news video, in combination with text analysis and a studio anchor-person detector.

By extracting a single frame or *key frame* from each detected video shot, one can effectively transform a collection of video files into a still image collection and apply content-based image retrieval techniques. We present our work in which global features were extracted from key frames and used as the basis for content-based retrieval using a vector space model, k-nearest neighbours and boosting. We report on an extensive comparison and evaluation of these techniques.

Further information can be extracted from key frames by looking at individual *regions* within the images. We describe our work in classifying regions of key frames, facilitating the labelling of video shots with appropriate semantic categories.

Finally, we have developed a news retrieval and summarisation system which automatically identifies stories in news broadcasts and uses key frames and text summarisation techniques to provide a summary of each story. We describe how this system has been integrated with a browsing framework that facilitates content-based video retrieval by building on our experiments using global features from key frames.

Acknowledgements

"You are worthy, our Lord and God,
to receive glory and honour and power,
for you created all things,
and by your will they were created and have their being."
— *Revelation 4:11*

My thanks ...

- to my supervisor, Stefan Rüger, for inspiration, wise advice and encouragement.
- to David Sinclair, for valuable ideas and for supervising my internship at AT&T.
- to Lawrence Wong, who did much of the early implementation work on the ANSES interface and infrastructure.
- to Daniel, Shyamala, Alexei and Peter, for being helpful and interesting group colleagues, and for useful collaborations.
- to the EPSRC, AT&T Laboratories and the Multimedia Knowledge Management Group, for financial support.
- to Mum and Dad and all the family, for giving me the best start in life and for giving me the opportunities that have got me to where I am.
- to all my friends at Holy Trinity Brompton, for fun and laughter, for never-ending support and for helping make sure there's been so much more to life than study.

Contents

Chapter 1

Introduction

1.1 Motivation

Since the advent of television in the 1920s, organisations like the BBC have been accumulating large archives of broadcast video data. As the cost of digital storage has fallen and network technology has improved, these archives are increasingly being digitised and made available to wider audiences. These advances have brought with them a demand for effective video retrieval solutions.

Vast numbers of text-based documents have become available on the Internet in the last few years. Information on virtually every subject is available, and the Internet has become an important medium for the communication and sharing of material that was otherwise difficult or impossible to obtain. Many newspapers now offer online editions that allow for easy perusal of the day's news, and search facilities for finding articles from previous editions. Hyperlinked pages allow for easy cross-referencing and acquisition of additional material on a subject. It is envisaged that television broadcasts, such as news programmes, could be made available in a similar way, making the viewing experience more akin to that of browsing a magazine, where irrelevant or uninteresting articles can be skipped, and others browsed in more detail.

A certain amount of useful information can, of course, be retrieved if the archive has been effectively catalogued, but what one is able to retrieve is determined by what the cataloguer wrote down, rather than by the actual content of the broadcast. This problem highlights a need for *content-based* information retrieval.

In our initial work [67] we partially solved this problem by the use of transcripts taken from subtitles, and others have derived transcripts by using speech recognition [105]. Capturing the subtitles broadcast with a TV programme provided the basis for a keyword search to be carried out. Effective though this method proves to be for news searching, it takes no account of the visual content of the broadcast and therefore still places a restriction on the type of search that can be performed on the data.

1.2 Thesis overview

The aim, then, of this thesis was to build on our earlier work by adding more content-based retrieval functionality. This involved looking at ways of summarising video by integrating information taken from transcripts generated from speech recognition and subtitles, with other information derived from the graphical content. The bulk of the work falls into three broad categories:

- **Boundary detection.** Boundary detection determines the fundamental units of retrieval (Chapter 3).
- **Key frame analysis.** Key frames are derived from the boundary detection process. The key frames for a broadcast can then be treated as a database of still images, to which image retrieval techniques can be applied. We have completed work on the use of feature vectors (Chapter 4) and colour classifiers (Chapter 5).
- **Text analysis.** The use of text transcripts derived from subtitles and from automatic speech recognition has been shown to be an effective aid in the story boundary detection process (Chapter 3), in keyword-based retrieval and in automatically generating summaries of video clips (Chapter 6).

These three components are integrated into a video retrieval, summarisation and browsing system described in Chapter 6.

Chapter 2

Literature Survey

2.1 Introduction

We outline the literature in the field of video retrieval and summarisation. In Section 2.2 we describe work in the area of boundary detection. Video retrieval is outlined in Section 2.3 and summarisation techniques are outlined in Section 2.4. Finally, in Section 2.5 we describe common performance measures used in information retrieval, including those that we use in our later evaluations.

2.2 Video segmentation

Since a single broadcast may be anything up to several hours in length, it is necessary to define a smaller unit for retrieval. Users will usually want to pinpoint a location within a video, rather than simply retrieving the whole broadcast. In a networked environment like the internet there are also bandwidth considerations. A half hour broadcast in MPEG-1 format will take several hundred megabytes, and even with a broadband connection this will take a significant time to download.

So, one of the most fundamental tasks in a video retrieval system is some kind of segmentation, and this is usually achieved either through *shot* boundary detection, or *scene* boundary detection. A shot is a single, continuous camera action[51] — a physical unit of retrieval, while a scene (or story) is a more semantic notion — essentially a story unit. This structural hierarchy is shown in Figure 2.1. Shot boundary detection depends on low-level image features, but scene boundary detection is a much harder problem, requiring high-level

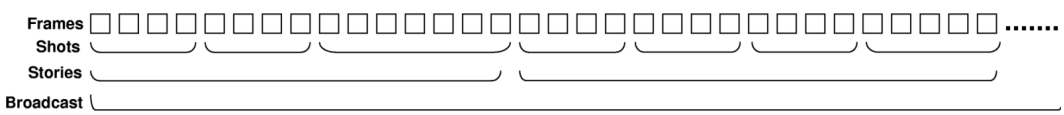

Figure 2.1: Structure of a video broadcast.

Figure 2.2: Illustration of a shot boundary made by a simple hard cut.

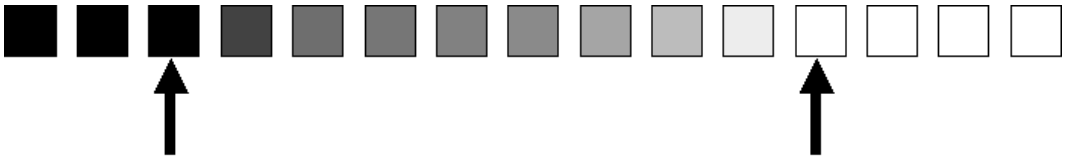

Figure 2.3: Illustration of a gradual shot change. The arrows indicate the start and end of the transition — ideally one would compare these two frames to detect a transition.

reasoning. We describe our own work on video boundary detection in Chapter 3, but some of the existing work is described here.

2.2.1 Shot boundary detection

Shot boundary detection essentially involves examining the information contained in individual video frames and comparing this with other nearby frames to determine if a shot change has taken place. A number of algorithms have been proposed in recent years in an attempt to solve the problem. Initially, these would look at differences between consecutive frames to determine whether any significant change had occurred. This works well for hard cuts — where a shot ends on one frame, and the next shot begins cleanly on the next frame, as shown in Figure 2.2 — but less so for gradual transitions where the new shot is introduced over a number of frames (see Figure 2.3). Simple consecutive frame comparison methods are first described, before a discussion of more complex algorithms dealing with all types of transitions.

Frame comparison methods

Pairwise pixel comparison. Perhaps the most intuitive way of determining whether a pair of frames represents a shot boundary is to compare corresponding pixels in the two frames. In pair-wise comparison [115], the idea is to compare two frames and decide how many pixels have changed. A pixel is judged as having changed if the difference in intensity between the two frames is greater than a pre-defined threshold. A shot boundary is then declared if a pre-determined percentage of pixels have changed. Pixel by pixel comparison has been shown to give some good results [6], but this algorithm is particularly sensitive to camera motion; a camera pan of even a few pixels across a rapidly varying scene could cause every pixel to be declared as changed, resulting in declaration of a false positive.

Likelihood ratio. In order to circumvent the problem of small camera motions falsely triggering shot changes, Kasturi and Jain [43] suggested comparing regions of an image, rather than individual pixels. For each pair of corresponding regions, they calculated a statistical value — the *likelihood ratio* — and if this exceeded a threshold, the region was said to have changed. Then, if enough *regions* had changed, a shot change was declared.

Histogram comparison. Rather than comparing corresponding pixels or regions, it is possible to use a feature of the entire image upon which to base the shot change detection decision. A histogram is a representation of the frequency with which pixels in an image fall into each of a fixed number of pre-determined quantisation levels. A histogram is computed for each frame in the video, and comparisons can be made between histograms, rather than between pixel values. Because of its balance of simplicity and effectiveness, histogram comparison forms the basis of many shot boundary detection algorithms that can be found in the literature [73, 90, 115], and works on the premise that two frames with unchanging backgrounds and unchanging objects will have little difference in their respective histograms of intensity.

In more detail, the basic computation is carried out as follows [115]: The histogram value for the ith frame is denoted by $H_i(j)$, where j is one of the G possible grey levels. G is determined based on the number of grey-levels in the original picture, and on the desired computation time. The difference between the ith frame and the following frame is then given by:

$$SD_i = \sum_{j=1}^{G} |H_i(j) - H_{i+1}(j)|$$

This value would be normalised by the number of pixels in the image if the algorithm was to be used in an environment where differing frame sizes were to be encountered.

There are a number of variations to this basic approach. Some approaches (Pye et al [73], for example) have used *colour* histogram comparison, where the individual colour channels are used, rather than overall grey-level intensity. Another common variation is to split the image into a number of blocks, and to do histogram comparisons between corresponding blocks. The median block difference is then used to determine whether or not a shot change has taken place. The advantage of this is that if there is significant local motion, affecting just one or two blocks, it is effectively ignored by the median calculation.

These algorithms face a number of limitations, most of which are caused by the fact that there are reasons other than shot boundaries for differences between frames. Firstly, if there is significant motion, either of a large (relative to the frame size) object, or at high speed, this is likely to cause a significant difference between frames. Changes in illumination cause problems — this is particularly common in news video, where flash-bulbs from other cameras cause a single frame to change its intensity completely. All methods are sensitive to rapid camera motion, which may give the effect of a shot change having taken place if the difference between two frames is large enough. Finally, none of the algorithms are able to

reliably detect *gradual transitions*, which are an integral feature of modern video production. Gradual transitions, such as fades, are effects which take place across a number of frames, and the difference between two consecutive frames may not, therefore, be significant enough to be detected (and if it was, then a whole series of shot boundaries would be registered — one for each pair of frames involved in the transition).

Handling gradual transitions

Extensions to histogram comparison. The basic histogram comparison method can be extended for gradual transitions by looking at histogram differences across a greater number of frames than just consecutive ones (see Figure 2.3). Zhang's twin comparison method [115] uses two thresholds. If the frame difference exceeds the higher threshold, a change is declared as before. If the frame difference only exceeds the lower threshold, an *accumulated comparison* is started, in which successive frame differences are then added to an accumulator. If, the accumulated difference exceeds the higher threshold before the consecutive difference falls below the lower threshold again then a gradual transition is declared. Our own algorithm [70, 68] extends the algorithm of Pye et al [73] by using enhanced peak detection and a method inspired by Zhang [115] for detecting the start and end of the shot transitions. Our work is described in more detail in Chapter 3.

Edge detection. Zabih et al [113] approach the problem of detecting different types of transitions by performing edge detection on each frame. They are able to make deductions about whether changes are cuts or certain types of gradual transition by analysing the properties and distribution of entering and exiting edge pixels. This algorithm was shown to be very effective, but is computationally very costly.

Compressed domain algorithms. A number of algorithms have been proposed to exploit the information already contained in compression schemes such as MPEG [47]. These schemes typically look at discrete cosine transform (DCT) coefficients and motion vectors. DCT coefficients can be compared between frames in a similar way to pixel matching [51]. In general, motion vectors exhibit relatively continuous changes within a camera shot, while the continuity would be disrupted between frames across different shots [40]. Mandal et al [51] have summarised other algorithms working in the compressed domain.

In our work we do not use compressed domain algorithms since they are format specific and it was thought necessary to have an algorithm that can work with all formats. The algorithms described above which work on raw pixel data are universal because all video data can be reduced to the basic pixel form. However, groups participating in TRECVID (described in the next section) that used compressed domain algorithms reported accuracy comparable to that for groups working in the uncompressed domain, with vastly reduced execution times. De-compression of MPEG data is the main processing time overhead in our shot boundary detection algorithm; a step which is unnecessary when working directly with the compressed data.

TRECVID 2003 shot boundary detection

The shot boundary detection task in the TREC 2001 and 2002 video tracks and the TRECVID 2003 workshop provided a useful forum for the comparison of state of the art shot boundary detection techniques. TREC is outlined in Section 2.5.3, and the shot boundary detection task itself is described in more detail in Section 3.2, but here we summarise notable entries.

The most successful system was designed by IBM [2]. Their system was based around a Finite State Machine, and looked at differences in RGB colour histograms, edge intensity histograms and thumbnails over distances of 1, 3, 5, 7, 9 and 13 frames. Adaptive thresholds were used, taking into account difference statistics in windows of 61 frames. Looking at the errors from their previous year's system, they added modules to handle photographic flashes, detection of fade out-in and errors in the MPEG stream.

The effective system developed by CLIPS-IMAG [74] was a modular system with components to deal with specific tasks and problems. Cut detection was performed by direct image comparison following motion compensation, and dissolves were then detected by comparing norms of the first and second temporal derivatives of the frames. A module was added for detecting photographic flashes (so that these could be excluded from the cut detection) and a further module for detecting cuts through use of a motion peak detector. Our own system, described in detail in Section 3.2 was the third most effective system overall.

Some groups took advantage of the fact that the data was MPEG-encoded, and worked in the compressed domain — including Accenture [38], who carried out a chi-squared test between 3 different histograms (global intensity, row intensity and column intensity) of consecutive I-frames. Post processing was then carried out to determine the exact location between I frames of the shot boundary.

The vast majority of approaches use colour histograms in some form [108, 17, 53, 100, 54, 114, 23]. Eichmann et al [23] combined histogram features with the edge detection technique proposed by Zabih et al [113] and a frame colour distance based on the cosine vector distance between frame thumbnails. Volkmer et al [100] took an image query by example approach in order to compare the current frame with windows on either side of it in order to determine whether a shot boundary had occurred. Shapes of the difference peaks were examined in order to determine the type of boundary, similar to an approach taken by Mas and Fernandez [53].

The TRECVID 2003 shot boundary detection task was an effective forum for the comparison of state-of-the-art shot boundary detection techniques. The precision and recall measures provided a comparison of the accuracy of the algorithms used by participating groups. It would, however, have been useful to have included some other measures and comparisons. Firstly, no indication was given of the execution time of the various algorithms. Algorithms working in the compressed domain were likely to have been far more efficient in terms of the processing time required to execute them, in comparison to algorithms working on raw pixel data, which are likely have had execution time close to real time. Secondly, it would have been interesting to compare the strengths and weaknesses of the different

algorithms, in order to see whether it was always the same types of transitions which caused problems, so that either the strengths could be combined or so that future research could be focused on the weak areas.

2.2.2 Scene and story boundary detection

Efforts in scene boundary detection usually take the approach of attempting to cluster shots which have already been determined by a shot boundary detection algorithm.

A general algorithm has been proposed by Yeung et al [111], in which shots are considered to be part of the same scene if they are visually similar and temporally close.

Hauptmann et al [32] suggested integration of various sources of information to determine if a story boundary had occurred in news video. They looked at various image features to see if a commercial break had occurred; they used information from the audio track (longer silences indicated significant changes) and information from the subtitles.

Pye et al [73] demonstrated that where video and audio breaks in a broadcast coincide, there is good evidence for the occurrence of a significant event. Breaks detected in the audio stream which correspond to breaks in the video are taken as strong indicators of scene boundaries. Once a video shot break has been detected using a shot boundary detection algorithm, the audio data in its neighbourhood is examined for evidence that might corroborate the video break. Pye et al used a speaker change detection method in which changes in acoustic characteristics are examined, and a boundary was declared at peaks in this measure.

TRECVID 2003 story boundary detection

Since the TRECVID 2003 corpus consisted mainly of news data, a story boundary detection task was introduced.

The most effective system in the evaluation was presented by Chaisorn et al from the National University of Singapore [11], and was an enhancement of their existing system [10]. The system works on the shot level and on the story level. At the shot level, shots are modelled using high-level object-based features (face, video text, shot type), temporal features (background change, speaker change, motion, audio type, shot duration) and a low-level feature (colour histograms). A decision tree is then used to classify the shot into one of a number of pre-defined genres. Hidden Markov Model analysis is then used to detect story boundaries according to the genre types and time-dependent features based on speaker change, scene change and cue phrases ("Good evening", "CNN New York", etc).

The group from Dublin City University [9] made the assumption that the occurrence of an anchorperson indicates the start of a new news story. They used a clustering algorithm to group similar shots in a broadcast, based on the similarity of their colour composition and the time between them. They then used three conditions to determine which of the clusters were anchorperson groups: i) Temporal range of shots higher than a threshold, since anchorperson shots are usually spread through the broadcast, rather than grouped

together. ii) Intra cluster similarity higher than a threshold, since the anchorperson shots should all be very similar to each other. iii) Mean shot length longer than a threshold, since anchorperson shots tend to be longer than other shots. Other indicators that they used were a face detector (since anchorperson shots always contain a face) and an activity measure (anchorperson shots are relatively static). This evidence was combined with an Automatic Speech Recognition (ASR) based analysis provided by StreamSage (see below) in a Support Vector Machine trained using a subset of the TRECVID development set.

Fudan University [108] took a similar approach to anchorperson detection and declared story boundaries at the change from a non-anchorperson shot to an anchorperson shot, and at the change from a commercial shot to a non-commercial shot. They also made use of the ASR text, and examined word histograms before and after shot and sentence boundaries.

IBM [2] used a Maximum Entropy statistical model to fuse multiple features such as motion, faces, music and speech, and text from the ASR stream.

StreamSage [77] took three approaches based solely on the ASR, which were combined to produce the final output: i) Noun-link: sentences containing repeated nouns are linked and segment boundaries are marked in places which break no links. ii) Boc-Choi: a similarity measure is computed over sentences and a similarity matrix formed over a moving window of five sentences. iii) Induced N-grams: Cue phrases are identified inductively by using the noun-link and Boc-Choi algorithms to determine candidate boundaries, and then listing all 1- 2- and 3-grams found near these boundaries as potential markers.

The University of Iowa group [23] also made use of cue phrases. In addition to this they looked for speech pauses longer than a threshold, on the assumption that they were good indicators of story boundaries. This evidence was combined with shot boundary detection.

The University of Central Florida [114] took a simple approach of detecting blank breaks between stories.

2.3 Retrieval

Given a set of shots or scenes delineated by the boundary detection process, we now turn to the problem of how to actually search for these clips — the proverbial "needle in a haystack" problem.

2.3.1 Keyword search

Traditionally, large archives of any kind of resource are managed by some form of cataloguing, and currently this is the way that commercial video archives are handled. At the BBC, video footage is manually annotated with information about the content, camera work and copyright information — a process which typically takes 30 person hours for each hour of video. Before computers, this was all handled with paper catalogues, in much the same way as a public library would be run. The increasing availability of computers brought about the obvious first step of computerised catalogues, meaning quick and easy keyword searches

through all the stored data. While this was a massive step in itself, it obviously does not circumvent the laborious manual cataloguing process. In the early days, when output was minimal, this manual cataloguing was a feasible process, but today, with the multitude of digital and satellite channels, it is becoming an increasingly impossible task.

The use of subtitles and speech recognition transcripts [8, 67, 56] has proved to be a stepping stone between totally manually generated catalogue keywords and a fully automated system for cataloguing and retrieval (which is yet to be realised). The basic idea is to associate with each shot keywords taken from the subtitles or from speech recognition transcripts, and to index these keywords. A simple text-based search engine interface can then be used for the user to enter keywords for a search.

Even as search methods progress beyond retrieval via transcripts, keyword retrieval is likely to remain important. With many of the retrieval methods described later in this chapter, there needs to be some language through which the user can enter the search terms, so it is likely that keywords will automatically be generated from whatever features are used as the basis for indexing. Town and Sinclair [96] have developed OQUEL, an image query language, which aims to allow specification of retrieval requirements through use of familiar natural language words.

2.3.2 Low-level feature extraction

Transcripts derived from the audio component of a broadcast obviously do not provide a full description of the broadcast. In much video material there is no dialogue — home movie footage, or music video for example. Whether or not dialogue and an associated transcript exists, user searches are often for a visual feature that is not described in the dialogue. See Chapter 4 for a fuller explanation of this problem. Research is now focused on ways of automatically extracting information from a number of important features of the visual component of video broadcasts.

Text

Text descriptions and transcriptions in the form of meta-data have already been described as an important aspect of video retrieval — but often text and captions also form part of the actual visual component of a broadcast, appearing embedded in the video stream and giving what is sometimes crucial information as to who is appearing or the subject matter. Some researchers have used optical character recognition (OCR) techniques to find and extract text from the video [80]. Successful groups participating in the search task of the TREC video track and TRECVID workshop claimed video OCR as an important part of their strategy [30].

Colour

Smith and Chang [88] describe VisualSEEk, in which queries are specified through colour and spatial layout of regions. Colour histograms are used for comparisons in a similar way to shot

boundary detection. Several groups participating in the feature detection task at TRECVID used colour histograms as a component of their feature detectors [38, 30, 108, 114, 2].

Rautiainen et al [76] add a temporal component to colour in the use of a Temporal Colour Correlogram, in which the correlation of colour pixels is captured through a sequence of video frames, rather than just within a single frame.

Our own work, presented in Chapter 5, was based on the work of Town and Sinclair [95], and aims to classify regions of an image into one of a number of visual categories. This would allow users to perform a keyword search for shots containing particular types of "stuff", for example grass or sky.

Texture

It has been shown [37] that texture-based features can perform better than simple colour features in image and video retrieval.

Camera motion

Users searching for particular video clips are often interested in specifying the type of camera action in the shot — for example, a shot in which the camera zooms in to a player of interest on a football field. Researchers have approached this by, for example, analysing the dominant motion in the shot [7].

2.3.3 Higher level semantic description

The low-level features described are often combined and used as the basis for higher level semantic description.

Object detection

Effective detectors for specific objects in specific domains can often be designed by making simple observations about colour and structure. Our own news anchorperson detector is described in detail in Section 3.3. Groups tackling the "weather news" feature at TRECVID typically used colour histogram models specific to the news channel [38, 108, 114].

Some specific object detectors have been generalised to work in a variety of contexts. Fleck et al [25] designed an algorithm for detecting naked people by looking for body parts based on colour and examining their structure. This work was generalised for other types of animals [27].

Much video material contains human beings, and these are likely to be central to many queries. Users will often want to find news footage about a particular person, or films containing a certain actor. There are two stages to the task: firstly, detecting the presence of the face in the video and, secondly, identifying it. Chellappa et al [13] presented a survey of techniques used in face recognition.

Schneiderman and Kanade [82] designed an approach to face and object detection that was popular amongst TRECVID participants and which used an AdaBoost classifier on a series of wavelet transforms.

Context

The TRECVID 2003 feature detection task included an "Outdoors" feature, in which participants had to detect scenes that were outside of buildings. Typically groups used colour based features and learning methods to train classifiers.

Szummer and Picard [91] combined colour, texture and frequency information from DCT coefficients to classify scenes as indoors or outdoors.

2.3.4 Motion

Representation of motion is unique to video, and thus it would seem to make sense to exploit this in some way. It can easily be seen that motion events could be useful in a query ("a fast moving car", "a tree falling") and Chang et al [12] designed a system in which users can make queries based on the temporal content of the video. A Java applet is provided for the user to sketch out arbitrary polygons and to specify a trajectory of motion for the described object. The system also allows for specification of arrival and death order of objects and their rate of growth. Each of these features is weighted according to its relative importance to the query, providing a basis for the ranking of returned results.

2.3.5 Audio

Much useful information can be gleaned from the soundtrack of a video — we have already mentioned the use of speech recognition transcripts. Informedia [31] also makes use of speaker identification techniques to determine who is speaking. Modelling techniques can also be used to help determine certain events that are taking place (an explosion, for example).

One of the features in the TRECVID feature detection task in 2003 was "female speech" and groups mainly used pitch-based analysis on the audio stream to perform this task [38, 108].

Other research has been carried out to discriminate between speech and music, since a user might specify this in a search (for example "I want a shot of a lunch on the lawn, with music playing in the background").

2.4 Interfaces

Even if one managed to design the world's best video retrieval system, it would be useless if there was no way of effectively presenting the results to the user for his or her perusal. A user must be able to quickly examine the retrieved results and make a decision as to whether they are relevant or not. This is complex with video, as if a large number of results were

returned it would be extremely time consuming to have to watch them all in order to make a relevance judgement. Therefore it becomes necessary to have some kind of summary of the content of a clip, and a number of concepts have been proposed and implemented.

2.4.1 Key frames

Almost universally, some form of *key frame* is used; a frame which is designed to represent all or part of a video shot.

A simple method is to use the nth frame (the first frame, for example) of each shot as the representative frame. If accurate shot boundary detection has been employed then in many cases a single frame will provide an accurate representation of the shot, but in some cases this will not be sufficient. For example, a camera operator might zoom in on an object of interest from a great distance and it may be many frames before the object is distinguishable; yet if only the first frame of the shot is used, the object of interest may be missed from the indexing process altogether. Also there is no clue as to the *action* of a shot. Some research is therefore centred on detecting camera actions within a shot, in order that relevant key frames can be selected.

Zhang et al [116] state that the challenge in key frame extraction is that it is automatic, content-based and represents the dynamic content of the video while removing redundancy, and has proposed a method whereby several key frames are extracted for each shot. Three criteria are defined for selecting key frames: *shot based*, *colour feature based*, and *motion based*. The *shot based* criterion states that at least one key frame will be selected for a segment, and this is always the first frame. Whether there is a need for other key frames for a segment is determined by the other two criteria. The *colour-feature based* criterion states that frames in a segment are processed sequentially and compared to the last chosen key frame. If a significant content change has occurred between the current frame and the previous key frame then the current frame is selected as a new key frame. The *motion based* criterion is motivated by the fact that significant motion in a segment is likely to represent important content change, since camera movement explicitly expresses the camera operator's intention in focusing on a particular object. Key frames are added according to types of motion that are classified as 'panning-like' or 'zooming-like'.

Mahindroo et al [50] employ object segmentation and tracking and select key frames based on critical events, such as appearance and disappearance of objects. Huet et al [39] consider groups of videos of a common type (news, soap operas etc) and look for shots that are specific to a particular episode, aiming to remove redundant shots. The summary then consists of one key frame for each shot that is considered significant. The work of DeMenthon et al [21] also focuses on finding important key frames, with the assumption that predictable frames are less important than unpredictable ones, for which clues often come from camera techniques employed.

2.4.2 Representative frames

Flickner et al [26] proposed a method in which the whole shot is represented in a single frame. For each shot, a representative frame ('r-frame') is generated. For static shots, this will just be a single frame from the shot, but for shots involving motion, a 'synthesised r-frame' is generated by mosaicking all of the frames in a given shot to give a depiction of the background captured in the shot. Foreground objects are then superimposed on the shot.

Arman et al [3] also propose a representative frame ('Rframe'). The body of the frame contains a shot from the video sequence (they choose the tenth frame), four motion tracking regions and shot length indicators. The motion tracking regions trace the motion of boundary pixels through time, acting as guides to camera or global motion. The time indicators provide an 'at a glance' view of shot length. The interface is not particularly intuitive, and it is not easy to see what is represented in each of the Rframes.

2.4.3 Text summary

The Informedia project at Carnegie Mellon University makes use of text streams from subtitles and automatic speech recognition transcripts to provide extra information about the video [15]. In news broadcasts, the audio stream (and therefore the subtitles and speech transcripts) are a rich source of information, as most of the important content is spoken. The combination of pictures and captions has been shown to aid recall and comprehension [46], as well as navigation [16], of video material and Informedia exploits this advantage by assembling 'collages' of images, text and other information (maps, for example) sourced via references from the text.

2.4.4 Other representations

Recent research in video summarisation has attempted to exploit the dynamic content of video, and rather than simply showing a static representation of the video, the aim is to discover relevant sections of the video and to play them to the user, so that the preview is a shortened version of the original. Ng and Lyu's method [61] simply involves playing a few seconds of each shot, while others opt to discover the important shots, and play them in their entirety. Oh and Hua [63] propose a system in which the user marks out a number of important scenes in the video and these are used as keys to search for other similar scenes based on visual content. Li et al [48] have designed an advanced browser which gives the user the option to view summaries created using compressed video generated by removing audio pauses (and the corresponding video frames) and a time compression technique which increases the playback speed while maintaining the pitch of the audio. Their interface also allows for navigating through the video according to the shot boundaries.

The Informedia group have developed a system in which the results of a search can be combined with geographic information [14]. Geographic references are extracted from the transcript and any overlaid text and location information then obtained from a gazetteer.

This information then allows a map to be displayed as the video of interest is played. The system also facilitates spatial queries, in which a user can click an area of interest on a map and retrieve associated video clips.

Systems presented at TRECVID 2003 were mostly key frame based. Oulu's VIRE system [76] combined a temporal view (consecutive shot key frames across the top of the screen) with a similarity view (columns below the shot key frames holding key frames most similar to the shot key frames). Worring et al [107] attempted to use 2-dimensional distances on the screen in such a way as to represent the multi-dimensional colour histogram distances between key frames. This work is similar in spirit to our own work (see Chapter 6), though our system interface additionally encapsulates key frame relationships based on multiple features. Browne et al [9] designed a web-based system in which the query key frames were displayed in a panel on the left of the screen, while the ranked key frames for the results were displayed in the main panel on the right, along with the associated text from the ASR stream.

2.5 Retrieval performance measures

2.5.1 The concept of relevance

Relevance is the fundamental criterion for evaluating the effectiveness of information retrieval (IR) systems [81], but is a confusing and much argued concept. The generally accepted definition of relevance is based on whether a user chooses to accept or reject information retrieved from a retrieval system. However, Schamber [81] lists a number of characteristics of relevance which, from the human judgement perspective, make relevance a difficult concept to measure:

- Subjective — depending on human judgement, thus not an inherent characteristic of information or of a document.
- Cognitive — depending on human knowledge and perceptions.
- Situational — relating to individual users' information problems.
- Multidimensional — influenced by many factors.

Clearly then, there can be no absolute measure of relevance and this issue has perplexed researchers through the years as it makes the important problem of evaluation a difficult one.

In addition to this, even when one has settled on an acceptable definition of relevance, there still remains the task of measuring it. Evaluation would generally involve manually assessing the relevance of every document (text file, image, video etc) in the database and comparing the assessments with those returned by the retrieval system. We already know that this is a costly (and often infeasible) process, since this is the problem we are designing systems to solve! Section 2.5.3 describes the Text REtrieval Conference (TREC) series, which has attempted to go some way to solving this problem.

2.5.2 Measures of IR system performance

A number of measures have been designed for the reporting of IR system performance. Recall and precision are the most common relevance-based measures used in IR testing [81].

Recall, precision and the precision-recall graph

Recall is the proportion of relevant documents in a database that are actually retrieved by the IR system, so gives a measure of the coverage of a system. Computing recall for a very large database is problematic, because one must know the absolute number of relevant documents in a system. Since, for large databases, it is infeasible to examine every document manually, it is common practice to estimate the number of relevant documents. TREC uses a technique called pooling. Another method is to examine a random sample of the database and determine the proportion of relevant documents in it; this proportion is assumed to be the same for the overall database.

Precision is the proportion of relevant items retrieved out of all items retrieved, and so gives a measure of the accuracy of a system.

It is generally accepted that there is an inverse relationship between recall and precision; that is, that precision decreases as recall increases, meaning that in order to retrieve more relevant documents, one has to sacrifice accuracy. This trend is demonstrated in a *precision-recall graph*, a plot of precision against recall.

A precision-recall graph is produced from a table of precision values at a number of recall values across the full range 0 to 1. Examples can be found in Figures 3.3 - 3.5. Graphs are particularly useful for comparing systems, as is the case in these examples. Comparisons are often made in the following three recall ranges [103]: 0 to 0.2 (high precision), 0.2 to 0.8 (middle recall) and 0.8 to 1 (high recall). Best performance is indicated by curves closer to the top right hand corner of the graph.

The main criticism of recall and precision as performance measures is that they are based on binary judgements of "relevant" and "non-relevant", ignoring the complexity of relevance judgements. It takes no account of the time and effort required for a successful search. The validity of recall as a performance measure can also be questioned; often users do not need or want to find all relevant documents, but rather one good quality document may satisfy their needs. Finally, recall is based on a concept — relevance — which we have already described as highly problematic and therefore perhaps unreliable.

Average precision

Average precision is a single-valued measure, used particularly in TREC, which reflects a system's performance over all relevant documents, rewarding systems that highly rank relevant documents. It is the average of the precision values obtained after each relevant document has been retrieved. If a relevant document is not retrieved, its precision is assumed to be zero [103].

Other performance measures

Lancaster and Warner [45] suggest alternative measures for performance evaluation, though they are based on essentially the same data. They are concerned with more directly measuring the time and effort involved in retrieval for the user.

Cooper [18] introduced the idea of measuring retrieval system effectiveness based on how long it took a user to find the document they were looking for. Dunlop [22] presented an "Expected Search Duration" measure in which precision and recall graphs are supplemented by plotting the number of relevant documents the user wishes to retrieve against the number of documents they would have to view to encounter them. de Vries et al [20] extended this work with a particular focus on evaluating retrieval where there is no predefined unit of retrieval — especially useful for video.

In TRECVID, participants are required to submit the time taken for a search, as well as the actual retrieved results, in order to give some measure of the effort required to perform a search. Some groups (for example DCU [9]) carried out extensive user studies.

Other researchers have tried to focus on measuring user satisfaction, not only with results, but also in the actual process of using the system. Such measures would look at how easy a user found the system to use, and how much they enjoyed using it. Users feel more satisfied if they like a system, even if its results are inferior to a system that they do not feel as comfortable with. Again, this is very subjective and is usually done through some sort of questionnaire.

2.5.3 TREC and TRECVID

The Text REtrieval Conference (TREC [101, 102]) has taken place annually since 1992 and attempts to solve some of the problems discussed in this chapter. The conference has four goals [102]:

1. to encourage retrieval research based on large test collections;
2. to increase communication among industry, academia and government by creating an open forum for the exchange of research ideas;
3. to speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems;
4. to increase the availability of appropriate evaluation techniques for use by industry and academia, including the development of new evaluation techniques more applicable to current systems.

A number of topics are set, describing some information need. Each participant performs one or more runs of the topics on their system and the results are sent to the U.S. National Institute of Standards and Technology (NIST), who run the conference. The problem of determining the number of relevant documents in the collection to calculate recall is solved

by a technique called 'pooling', whereby the top (say) 100 documents returned by each run are contributed to the pool. The pool is then assumed to contain all of the relevant documents, and the NIST assessors will then go about making a binary decision for each document in the pool of 'relevant' or 'non-relevant'. Voorhees [101] has shown that although the relevance judgements may vary enormously between assessors, the relative effectiveness of different retrieval strategies is stable.

Video retrieval in the TREC framework

A video track was introduced to TREC for the first time in 2001 [87] and consisted of two main tasks: shot boundary determination and search. In 2002 [86] a high level feature detection task was added. The video track was spun out as a separate workshop (TRECVID) in 2003 [85]; broadcast news data was used for the first time and so a news story boundary detection task was added.

Collaboration of this kind is enormously beneficial. It provides a forum for the comparison of all manner of different approaches to retrieval problems and to discover their relative benefits, and provides individual research groups with a valuable opportunity for a thorough evaluation of their systems.

Our own participation in the various TREC and TRECVID tasks is described in later chapters.

Chapter 3

Video segmentation

3.1 Introduction

In this chapter, we describe our work in the area of segmentation. Firstly, in Section 3.2 we describe our basic shot boundary detection scheme, and evaluation of this algorithm in the TREC and TRECVID workshops. Then, in Section 3.3, we show the application of the algorithm to the problem of news story boundary detection.

3.2 Shot boundary detection

At its most basic level, video is divided into frames — still images that, when displayed in quick succession, give the illusion of motion. A video shot is a sequence of these frames produced by a single camera in a single action, and is a useful unit for decomposing a video broadcast. The task of shot boundary detection involves automatically locating the transitions between these shots in a whole video broadcast.

Since our shot boundary detection algorithm has undergone extensive evaluation through the TREC video track and TRECVID workshop, we describe it in this context. We participated in the shot boundary detection task in each of the three years of its existence to date, and our experiments and results in each of these three years are described in the following sections.

3.2.1 TREC and TRECVID shot boundary detection task

Background

The task was to detect transitions in a defined set of video broadcasts and to classify each transition as either a hard cut or a gradual transition (dissolve, fade to black etc). The data set was different each year, and each year's data is described in detail below.

Evaluation

Groups participating in the TREC and TRECVID evaluations were required to submit the output of their automatic systems to NIST, where software was used to compare the system output against a manual annotation of the data set.

The manually annotated reference data (the ground truth) was created by a student working at NIST, who watched all of the video material and noted down the start and end times of each of the shot boundary transitions.

Automatic system outputs submitted by participating groups were compared to the ground truth using a modified version of the protocol proposed for the OT10.3 Thematic Operation (Evaluation and Comparison of Video Shot Segmentation Methods) of the GT10 Working Group (Multimedia Indexing) of the ISIS Co-ordinated Research Project. The TREC version has the following features [87]:

- A gradual transition of less than 6 frames is treated as a cut.
- A submitted cut matches a reference cut if the latter falls entirely within the boundaries of the former, after the former has been extended 5 frames on each end.
- Gradual transitions match if the intersection is at least 0.333 of the longer and 0.499 of the shorter transition, the default values from the earlier ISIS scheme.

3.2.2 Data

In 2001 and 2002, the data for the task consisted of videos in MPEG-1 format, mainly documentary-type broadcasts, varying in age and playback quality. In 2001 there were 42 videos with a total run time of approximately 6 hours and in 2002 there were 18 videos with a total run time of approximately 5 hours. The 2003 TRECVID workshop used a collection consisting of recent ABC and CNN news data. For shot boundary detection 13 videos were used with a total run time of around 6 hours.

3.2.3 System

2001 system

Our shot boundary detection method [70] was inspired by the video component of the scene detection scheme described by Pye et al [73].

Each video frame is divided into 9 blocks, and for each block a histogram is determined for each of the RGB components. The Manhattan distance between corresponding component histograms for corresponding blocks in two frames is calculated, and the largest of the three is taken as the distance for that block. The distance between two frames is then taken as the median of the 9 block distances. This helps eliminate response to local motion.

A difference measure is defined as follows:

$$d_n(t) = \frac{1}{n} \sum_{i=0}^{n-1} D(t+i, t-n+i),$$

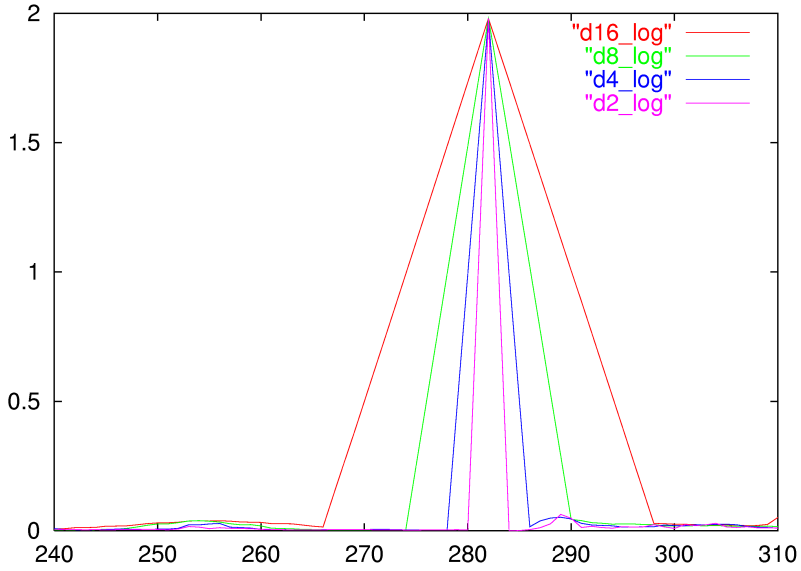

Figure 3.1: Characteristic d_n peak plot for a cut. The x-axis is time (frame numbers), the y-axis is the absolute value of d_n.

where $D(i, j)$ represents the median block distance between frames i and j.

A peak is defined as a value of d_n which is greater than a pre-defined threshold and is greater than the 16 preceding and 16 following values of d_n. A shot boundary is declared if there are near-coincident peaks of d_{16} and d_8. An additional coincident peak of d_2 suggests a cut, otherwise the boundary is classified as a gradual transition.

Figure 3.1 shows the typical pattern of peaks for a cut, where the coincident peaks of almost equal magnitude can be seen for all n. As Figure 3.2 shows, gradual transitions also have characteristic peaks, but are spread over a longer time period, and do not reach equal magnitude.

The TREC submission required declaration of the start and end times for gradual transitions, and so the algorithm described had to be extended to cater for this. The extension is similar to the method used by Zhang et al [115], in which a lower threshold is used to test for the start and end of a gradual transition. At each frame, the d_4 difference is compared to the threshold. If it is greater than the threshold it is marked as a potential start of a transition. If, on examination of successive frames, the d_4 difference falls below the threshold again before a shot boundary is detected, this potential start is scrapped and the search continues. Following the detection of a shot boundary, the end point of the transition is declared as the first point following the shot boundary at which the d_4 change falls below the threshold. The d_4 timescale is used because it is fine enough to accurately pinpoint the moment at which the change begins, but also introduces a tolerance to any momentary drop in the difference which may occur in the process of the change.

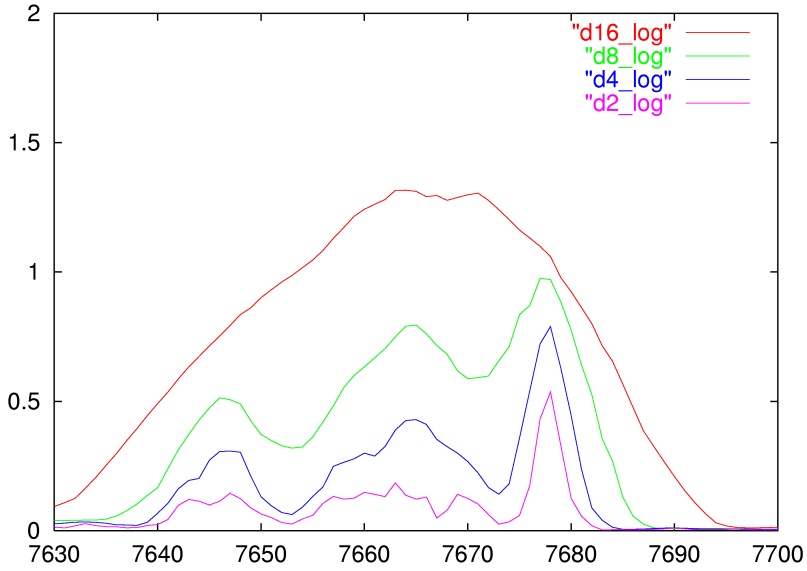

Figure 3.2: Characteristic d_n peak plot for a gradual transition. The x-axis is time (frame numbers), the y-axis is the absolute value of d_n.

2002 system

The shot boundary detection system used for the TREC 2002 video track [68] was fundamentally the same as the 2001 system, but empirical analysis of data from the original system led to the redefinition of the conditions for cuts and graduals, based on the characteristics of the peaks found at transitions of those types.

If, at frame f, the value for $d_{16}(f)$ is greater than an empirically determined threshold T_{16}, the frame is examined for the presence of a shot boundary. A cut was declared at frame f if the following empirically determined conditions held:

- $d_8(f) > T_8$ (where $T_8 = 0.4T_{16}$)
- $d_n(f) > d_n(f+\delta)$ for all $n \in \{2, 4, 8\}$ and all $\delta \in \{-2, -1, 1, 2\}$ (cuts show characteristic coincident peaks for all d_n).
- $d_n > 1.3d_{n/2}$

If no cut was declared, a gradual transition was declared if the following conditions held:

- $d_{16}(f) > d_{16}(f \pm \delta)$ for all $\delta \leq 16$
- Peak value of d_8 in range $f \pm 16$ occurs within $f \pm 5$

Six runs were entered with varying threshold settings (compared to the single run entered in 2001). The first three runs had a constant low (d_4) threshold which was believed to be

the optimum, and the high (d_{16}) threshold was varied. In the second three runs, the effect of varying the low threshold was examined while keeping the high threshold at a constant value that was believed to be optimum. Details of the values used are given in the results section.

It has been suggested that automatic threshold setting can improve performance [89]. We examined data plots from the 4 distance measures in regions where transitions were missed and found little evidence to support the value of adaptive thresholding. Missed transitions were usually connected with poorly defined peaks, and any threshold which caught these would also have caught nearby peaks caused by other occurrences such as camera motion.

2003 system

The system we employed for the TRECVID workshop in 2003 [34] was largely unchanged from the 2002 system, but results from 2002 were used to inform the threshold settings, and the more recent experiments are noteworthy due to the use of recent broadcast news data, as opposed to the ancient footage used in the previous two years.

We performed ten shot boundary detection runs in 2003, similar to those performed in 2002, but taking into account the weaknesses in the 2002 submission by performing more runs with a higher T_{16} threshold.

3.2.4 Results

TREC 2001

Results for each of the video broadcasts in the shot boundary test set are shown in Table 3.1 (cuts) and Table 3.2 (gradual transitions), which show that, compared with 14 other systems, our method performed better than average. This was in the absence of any fine tuning — there are a number of thresholds which were set based only on one training video.

TREC 2002

The TREC 2002 video track provided an opportunity to enhance our system and test on a larger corpus of material.

We performed six shot boundary detection runs. The first three runs, KM-01 — KM-03 were carried out keeping the low threshold, T_4, constant at 0.05, and setting the high threshold, T_{16}, at values of 0.5, 0.4 and 0.3 respectively. In runs KM-04 — KM-06, the low threshold was increased to 0.15, and the same 3 values for the high threshold were used again.

We show the results for our six shot boundary detection runs in Table 3.3. All six runs gave good results for overall precision and recall, comparing favourably with the average of all systems (shown as "Median" in Table 3.3). System KM-01 appeared to give the best balance between precision and recall overall, suggesting that further experiments with a higher T_{16} threshold may be worthwhile — therefore this was tried in 2003.

Video	Known	Recall		Precision	
	Trans	Mean	Ours	Mean	Ours
ahf1.mpg	63	0.961	0.936	0.919	0.921
ancc836y.mpg	2	0.875	0.500	0.889	1.000
anni005.mpg	38	0.929	0.973	0.699	0.770
anni009.mpg	38	0.870	0.789	0.735	0.697
bor03.mpg	230	0.856	0.982	0.920	0.953
bor08.mpg	379	0.920	0.873	0.894	0.945
bor17.mpg	127	0.905	0.952	0.843	0.975
eal1.mpg	61	0.956	0.967	0.943	0.921
ldoi874_2.mpg	8	0.896	1.000	0.776	1.000
ldoi874_4.mpg	7	0.888	0.714	0.688	0.625
ldoi909j.mpg	1	0.333	0.000	0.292	0.000
nad28.mpg	181	0.917	0.939	0.853	0.762
nad31.mpg	187	0.892	0.866	0.834	0.900
nad33.mpg	189	0.951	0.952	0.882	0.918
nad53.mpg	83	0.962	0.951	0.876	0.797
nad57.mpg	44	0.956	0.977	0.880	0.860
nbmw628d.mpg	1	1.000	1.000	0.958	1.000
pfm1.mpg	61	0.948	0.967	0.851	0.746
senses111.mpg	292	0.902	0.989	0.909	1.000
ydh1.mpg	69	0.961	0.971	0.839	0.881
Weighted mean		0.915	0.935	0.876	0.907

Table 3.1: TREC 2001 Video Track shot boundary detection task — performance on cuts in all files. The recall and precision values for our system are shown next to the respective means across all systems. The bottom line shows the column mean for each of the statistics, with each file's contribution weighted by the number of transitions in that file.

Video	Known	Recall		Precision	
	Trans	Mean	Ours	Mean	Ours
ahf1.mpg	44	0.683	0.681	0.700	0.625
anni005.mpg	27	0.609	0.444	0.679	0.666
anni009.mpg	65	0.501	0.523	0.669	0.809
bor03.mpg	11	0.660	0.545	0.283	0.166
bor08.mpg	151	0.633	0.741	0.794	0.741
bor10.mpg	150	0.687	0.866	0.743	0.866
bor12.mpg	136	0.556	0.625	0.705	0.825
bor17.mpg	119	0.511	0.697	0.678	0.584
eal1.mpg	20	0.573	0.600	0.531	0.600
nad28.mpg	116	0.603	0.543	0.555	0.588
nad31.mpg	55	0.478	0.418	0.436	0.353
nad33.mpg	26	0.535	0.500	0.389	0.206
nad53.mpg	76	0.596	0.631	0.575	0.761
nad57.mpg	23	0.659	0.826	0.620	0.612
pfm1.mpg	21	0.630	0.666	0.499	0.608
senses111.mpg	16	0.336	0.187	0.298	0.068
ydh1.mpg	52	0.492	0.423	0.620	0.536
Weighted mean		0.585	0.640	0.648	0.670

Table 3.2: TREC 2001 Video Track shot boundary detection task — performance on gradual transitions in all files. (Note that some of the files which appeared in the previous table contained no gradual transitions and are therefore not listed in this table)

	All		Cuts		Gradual			
	Rec	Prec	Rec	Prec	Rec	Prec	F-Rec	F-Prec
KM-01	0.826	0.843	0.883	0.895	0.682	0.707	0.673	0.608
KM-02	0.845	0.798	0.889	0.863	0.733	0.648	0.658	0.618
KM-03	0.859	0.720	0.893	0.803	0.773	0.553	0.650	0.612
KM-04	0.825	0.813	0.888	0.880	0.665	0.645	0.471	0.603
KM-05	0.833	0.755	0.891	0.832	0.685	0.578	0.477	0.444
KM-06	0.836	0.688	0.885	0.755	0.711	0.536	0.477	0.356
Median*	0.785	0.825	0.884	0.871	0.622	0.701	0.576	0.762

Table 3.3: TREC 2002 Video Track shot boundary detection task — summary of results for the six runs submitted. *This is the median of ALL runs submitted by ALL participants in the 2002 task (not the column median). Rec=Recall, Prec=Precision.

	T_{16}	T_4
Imperial-01	0.60	0.05
Imperial-02	0.50	0.05
Imperial-03	0.40	0.05
Imperial-04	0.30	0.05
Imperial-05	0.60	0.15
Imperial-06	0.50	0.15
Imperial-07	0.40	0.15
Imperial-08	0.30	0.15
Imperial-09	0.50	0.10
Imperial-10	0.40	0.10

Table 3.4: Threshold settings for the TRECVID 2003 shot boundary detection runs.

The frame-recall and frame-precision results (F-Recall and F-Prec respectively in Table 3.3) give an indication of the accuracy of the system for detection of gradual transitions. Our relative performance here was not as good, and this perhaps reflects the fact that little time was devoted to tuning the algorithm for setting the start and end times of gradual transitions. Fine pinpointing of these times has never been an important part of our research, since we have only ever needed a single arbitrary point in the middle of the transition at which to split the broadcast.

TRECVID 2003

The complete set of threshold settings for the ten runs is shown in Table 3.4.

The results achieved, shown in Table 3.5, were impressive. The comparison with other groups participating in TRECVID, graphed in Figures 3.3 — 3.5, clearly shows our system (represented by the character 'o') as the third best amongst all systems. The trade-off between precision and recall as the thresholds were adjusted can clearly be seen.

Little work had been done on tuning the algorithm for the broadcast news data for TRECVID 2003, and some empirical analysis of the type carried about before the TREC 2002 video track could make for a significant improvement for 2004, when data of a similar type will be used.

Although our shot boundary detection algorithm has been proven extremely accurate and has produced impressive results in TREC and TRECVID, it is worth noting that it was originally conceived when we had the limitation of working with uncompressed data captured in our BBC news processing system. MPEG has become a standard for storage of video data and the format contains a great deal of information that is useful in a shot boundary detection algorithm. If starting from scratch it would now seem a sensible approach to design a shot boundary detection system that would work in the compressed domain, given the huge advantage in execution time. Working in the uncompressed domain has the advantage that

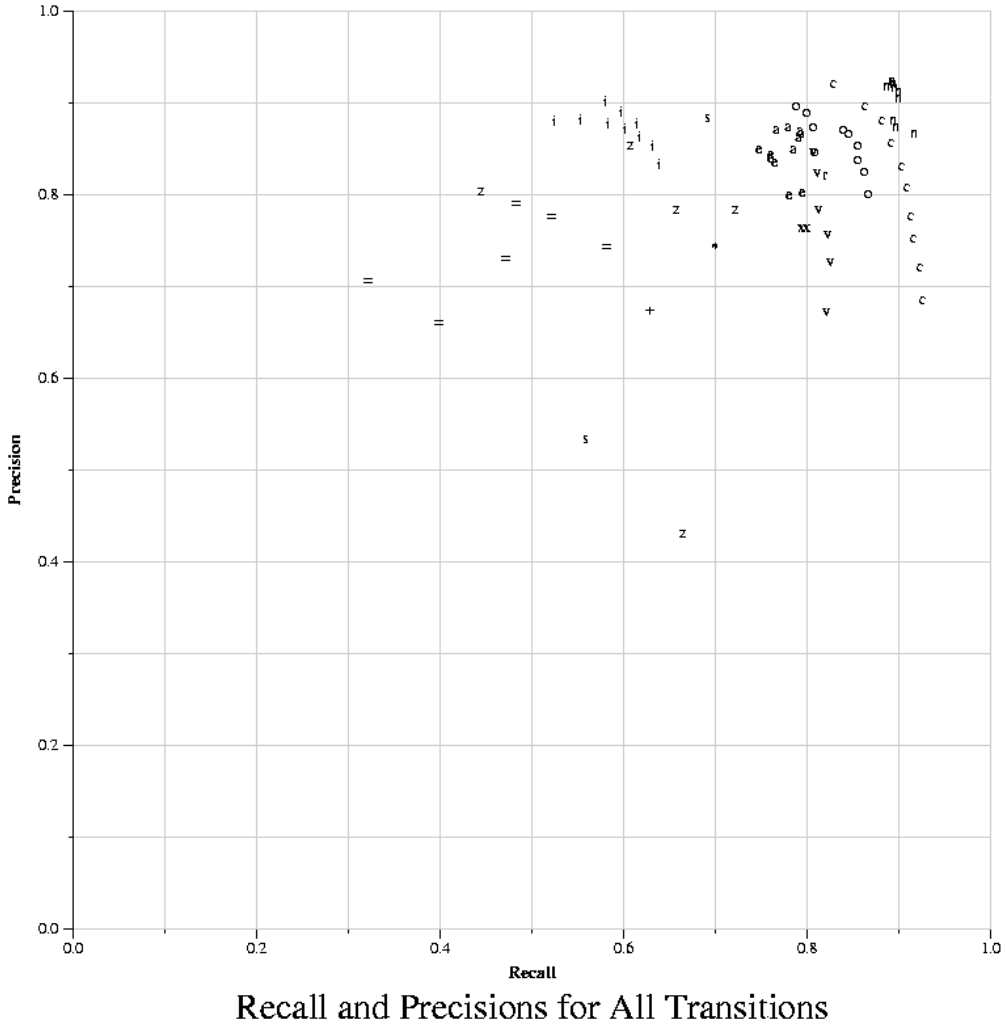

Figure 3.3: Average recall and precision results for all transitions for all participating shot boundary detection systems in TRECVID 2003. Our system is denoted by the character 'o'

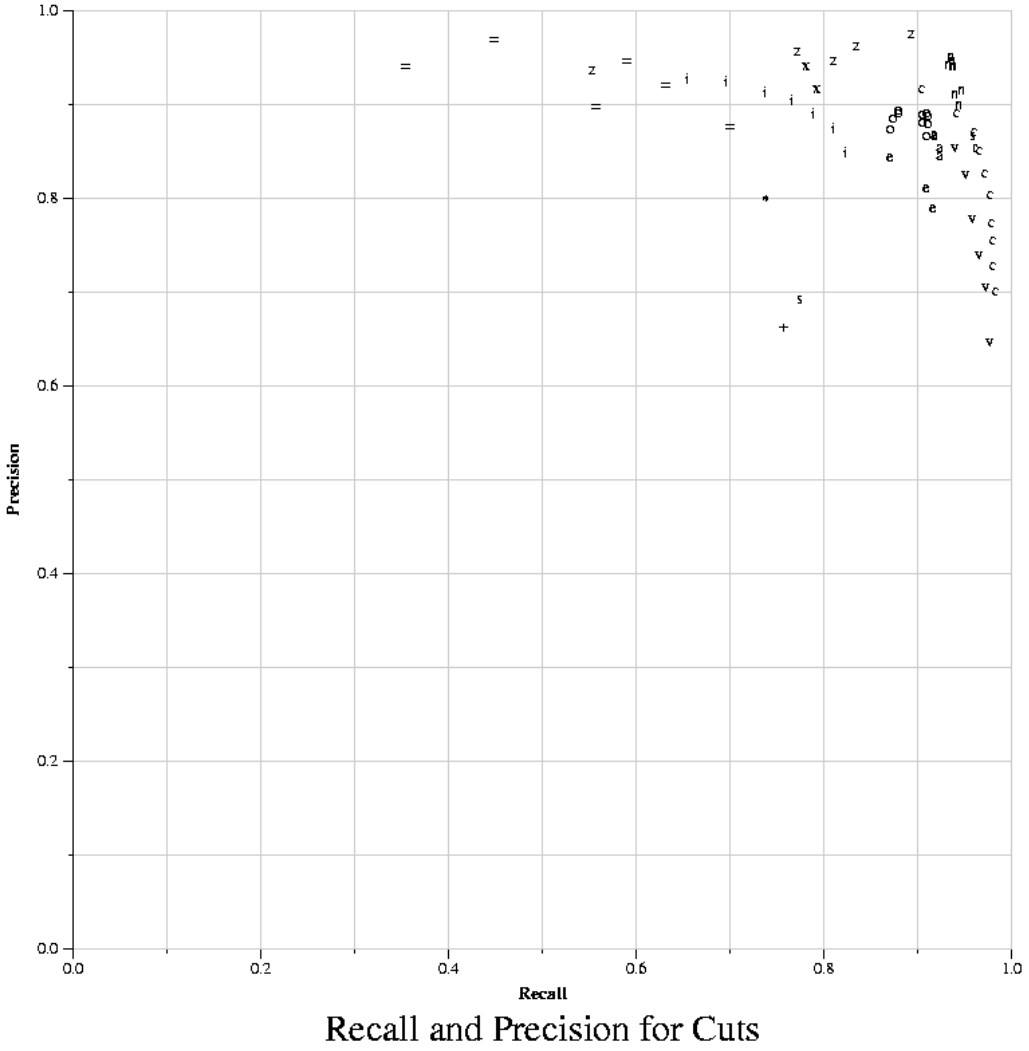

Figure 3.4: Average recall and precision results for cuts for all participating shot boundary detection systems in TRECVID 2003. Our system is denoted by the character 'o'

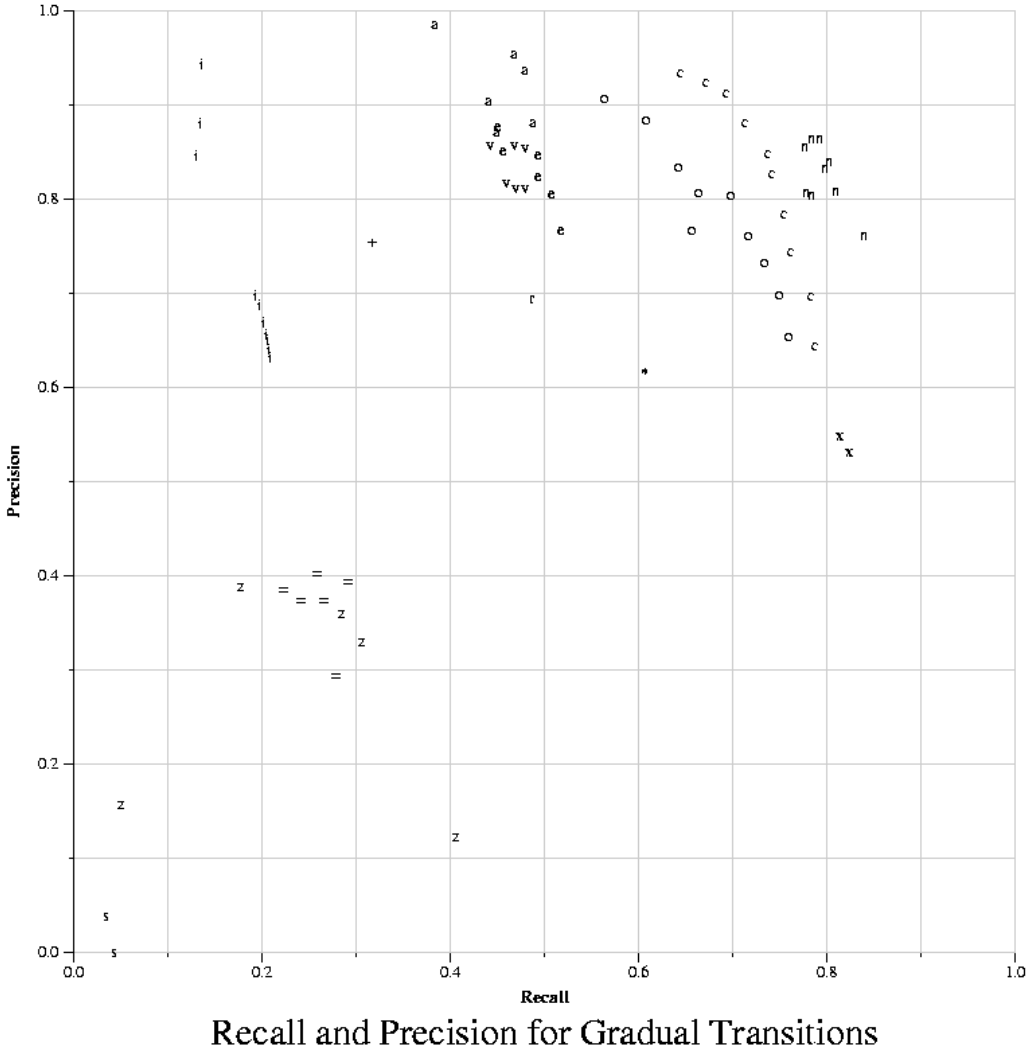

Figure 3.5: Average recall and precision results for gradual transitions for all participating shot boundary detection systems in TRECVID 2003. Our system is denoted by the character 'o'

	All		Cuts		Gradual			
	Rec	Prec	Rec	Prec	Rec	Prec	F-Rec	F-Prec
Imperial-01	0.788	0.897	0.880	0.895	0.564	0.907	0.799	0.239
Imperial-02	0.800	0.890	0.879	0.892	0.608	0.885	0.811	0.245
Imperial-03	0.807	0.874	0.874	0.886	0.643	0.834	0.819	0.256
Imperial-04	0.808	0.847	0.871	0.874	0.657	0.768	0.823	0.258
Imperial-05	0.839	0.871	0.910	0.892	0.664	0.808	0.704	0.627
Imperial-06	0.855	0.854	0.911	0.889	0.717	0.762	0.704	0.626
Imperial-07	0.863	0.826	0.911	0.880	0.749	0.699	0.698	0.628
Imperial-08	0.866	0.801	0.910	0.868	0.760	0.654	0.697	0.597
Imperial-09	0.845	0.868	0.905	0.890	0.698	0.805	0.757	0.477
Imperial-10	0.855	0.839	0.905	0.882	0.734	0.733	0.747	0.484
Median*	0.795	0.849	0.910	0.881	0.487	0.806	0.750	0.707

Table 3.5: TRECVID 2003 shot boundary detection task — summary of results for the ten runs submitted, compared with the median for all systems submitted by the various groups to TRECVID. Rec=Recall, Prec=Precision. *This is the median of all systems submitted by all groups, not a column median.

all formats can easily be uncompressed, but working with compressed data should not prove too much of a problem if one is designing an algorithm for a specific application.

3.3 Story boundary detection

Shot boundary detection is a highly important process in determining the structure of video material, but the highly fragmented nature of modern video production means that retrieving video on a shot by shot basis is likely to be a frustrating experience, and we therefore need to look to a higher semantic level. In films, for example, a useful retrieval unit might be a *scene* — a group of shots from a single location. In broadcast news material, one of the most useful units for retrieval is the *story*.

The TRECVID 2003 workshop included a story boundary detection task and, as with the shot boundary detection system, our work is described in this context since it provides a useful framework for evaluation. We did not formally participate in the story boundary detection task at the workshop itself, but evaluation results are generated using the TRECVID ground truth and evaluation software.

Since our story boundary detection system was designed for our BBC news retrieval system we also include anecdotal observations and examples from this system.

3.3.1 TRECVID story boundary detection task

Background

The task was to identify story boundaries in the test collection along with their times. There was also an optional task to identify the type of the "story" — miscellaneous or news — though we have not attempted classification of this type.

Evaluation

Evaluation of story boundary submissions was carried out in a similar way to that of the cuts in shot boundary detection. Story boundaries were declared as a time relative to the start of the video file. Each reference boundary was expanded by five seconds in each direction, giving an evaluation interval of 10 seconds, thus a submitted boundary would be declared correct if it fell within this evaluation interval, and a false alarm otherwise.

As with the shot boundary detection, performance was measured in terms of precision and recall.

3.3.2 Data

The story boundary test collection was much larger than the shot boundary test collection and consisted of approximately 60 hours of CNN and ABC news video, made up of approximately 115 broadcasts. The ground truth for the story boundaries was developed by LDC for the TDT project.

Text data for the news broadcast was provided in two forms. Firstly, automatic speech recognition (ASR) transcripts generated by LIMSI [29] and, secondly, subtitles provided by LDC. The ASR transcripts were complete with timing information, but contained many word errors (as one would expect for an automatic system such as this), and no punctuation or other sentence structure information. The subtitles were more accurate, containing punctuation sufficient to reconstruct sentences, but had no timing information and were simply stored as a list of tokens, one per line.

Our system also runs daily on the BBC news and at the time of writing a corpus of approximately 9 months' news had been built up, consisting of around 270 broadcasts of 35 minutes length each. No formal evaluation of the story boundary detection in this system has been carried out, but subjectively the output is more impressive than that produced for the ABC and CNN data. This may largely be due to the way in which the subtitles are broadcast in complete sentences with some implicit grouping into stories. This is discussed in further detail in Section 6.2.1. There may also be differences in the language models which have not been investigated.

3.3.3 System

In contrast to the mainly text-based methods deployed in the topic detection and tracking research field [1], we work under the assumption that story boundaries always coincide with

video shot boundaries, and that providing the shot boundary detector has not missed the boundary, the problem of story segmentation is simply one of merging shots.

As a consequence, our shot boundary detection system is common to all approaches to story boundary detection described below. The output of the shot boundary detector is the timing information for the start and end of the shot and a single key frame to represent the shot.

We have designed two basic approaches to story boundary detection that work on top of the shot boundary detector: a text-based approach using the subtitles or speech recognition transcripts and an approach using anchorperson shot detection. These two approaches were also combined, as described below.

All training of the systems was carried out on our BBC news corpus or on the TRECVID development set.

Text based story boundary detection

Text pre-processing. Our text-based story boundary detector was designed to work with complete sentences, and therefore some preliminary processing of the text data was necessary to render it in a suitable format. Since the subtitle data was much better quality, we decided to use this as the primary text data source, but use the timing information in the LIMSI transcript to align it.

The alignment was carried out as follows: Firstly, complete sentences were formed from the tokens in the LDC subtitle transcripts. This was done by working on the assumption that a full stop delimits a sentence — but we had to account for common exceptions where a full stop denotes an abbreviation, such as "mr." or "u.s.". Following sentence extraction, Marc Lehmann's Perl String::Similarity function [1] was used to find for each subtitle sentence the corresponding sentence in the LIMSI transcript. This was done by moving the subtitle sentence word by word across the entire LIMSI transcript and finding the highest similarity score given by String::Similarity. Because of the inconsistencies between the subtitle transcripts and the LIMSI transcripts there was always the possibility of matching on the wrong part of the transcript — especially with common phrases. To circumvent this, two weights were factored in. The first weight was based on the relative distance of the match through the two transcripts, so that if we have a phrase 25% of the way through the subtitle transcript, a similar phrase 25% of the way through the LIMSI transcript would be weighted more heavily than a phrase 90% of the way through. The second weight was designed to encourage preservation of the temporal ordering of the sentences, and weighted more heavily those matches found that were closer to the previous match — thus suppressing big gaps in the times. Finally, a smoothing stage was carried out to re-order any out of sequence timings.

Segment merging algorithm. The following function returning a score for the similarity of two text segments was used:

[1] available from http://search.cpan.org/~mlehmann/String-Similarity-1/

$$\text{Similarity} = \sum_m \frac{w(m)}{d(m)},$$

where $w(m)$ is the weight of word in match m according to its type and $d(m)$ is the number of subtitle lines between the two words in the match. Words, when matched, will generate a different score according to their type, as shown in Table 3.6. A word may have more than one type and scores are accumulated accordingly — for example 'Beijing' is both a location and a noun, and would score $40 + 10 = 50$. The 'Other words' category includes all non-stopped words that do not fit another category.

Word type	Score
Organisation	60
Person	60
First Person	40
Location	40
Date	30
Noun	10
Other words	5

Table 3.6: Empirically determined word-type weightings

Each text segment is compared to each of its 5 neighbours on either side. When the similarity of two segments is greater than an empirically defined threshold, the two segments are merged with each other, and with any other segments that lie temporally between them. The corresponding video shots are also then merged.

Addition of synonyms. In order to improve the possibilities for matching of text segments, we carried out two runs in which we increased each segment's vocabulary by introducing synonyms for all words using the WordNet database [55], and comparing words as above.

In the future it may be worth extending this to expand the geographical vocabulary by use of a gazetteer — for example, a segment with a mention of "Baghdad" is likely to be closely related to one with a mention of "Iraq". For high precision tasks such as question answering, it has been argued that the use of a gazetteer introduces more ambiguity than anything else, but it seems likely that it could be of help in this situation.

Anchorperson-based story boundary detection

In this method we work with the additional assumption that each news story starts with a studio anchorperson. Based on this assumption, we detect anchorperson shots and use the corresponding shot start time as the story start time.

The basis for the anchorperson detector is the k-NN based retrieval system described in Section 4.4. Separate training was done for CNN news and ABC news (since their

studios differ in presentation) as follows. All anchorperson shots were manually marked in 10 randomly selected broadcasts from the development set for each channel, and then 30 positive examples randomly selected from each of these subsets. 100 negative examples were randomly selected for each channel from the complement of each positive subset.

The k-NN method returns a ranking value for each key frame in the test collection, and it was necessary to set a threshold such that all key frames ranked above it would be classified as anchorperson shots. From the 20 manually marked broadcasts (10 for each channel) it was determined that on average 2% of key frames were anchorperson shots, and the initial threshold was set such that the k-NN method would classify a similar proportion of key frames from the test set as anchorperson shots.

Finally, for each shot output from the shot boundary detection process, if that shot was an anchorperson shot, we merged with it all following shots, up until the start of the next anchorperson shot.

Figures 3.6 and 3.7 show respectively the 30 positive examples used for the ABC news anchorperson detector and the top 49 shots detected by it. This is a visual demonstration of its effectiveness. The CNN detector shows similarly impressive results.

Combining methods

We have tested a combined system in which the text based system was run on the output of the anchorperson detector to improve its precision.

3.3.4 Results

Table 3.7 shows the recall and precision values achieved by our ten unofficial runs.

Initially the output of the shot boundary detector was run through the evaluation software without any further processing, in order to determine the "baseline" for further experiments; this is the run *shot-1*. Since all of our story boundary detection methods work on the output of the shot boundary detector, we cannot expect recall any higher than 0.899, but we would hope to improve on the precision.

Runs *text-1* and *text-2* are runs using the simple text-based segment merging algorithm on the aligned subtitles, and *limsi-1* and *limsi-2* using the segment merging algorithm on the LIMSI ASR transcripts. Thresholds were set based on values found to be optimum through experimenting with the development set. As the merge threshold is increased, less segments are merged. The results clearly show the superiority of the subtitle data over the LIMSI transcripts, confirming our decision to use the subtitles as the text source in all other experiments. The text-based segment merging algorithm clearly improves the precision of the detected story boundaries, relative to basic shot boundary detection, though there is also a marked decrease in recall, resulting from incorrect merges being made.

Runs *synonyms-1* and *synonyms-2* show the effect of adding synonyms to the word list for each segment. There is not an appreciable difference to the results — though a fairly simplistic approach was taken. In the future it would be interesting to try introducing lexical

Figure 3.6: Positive examples used to train the ABC news anchorperson detector.

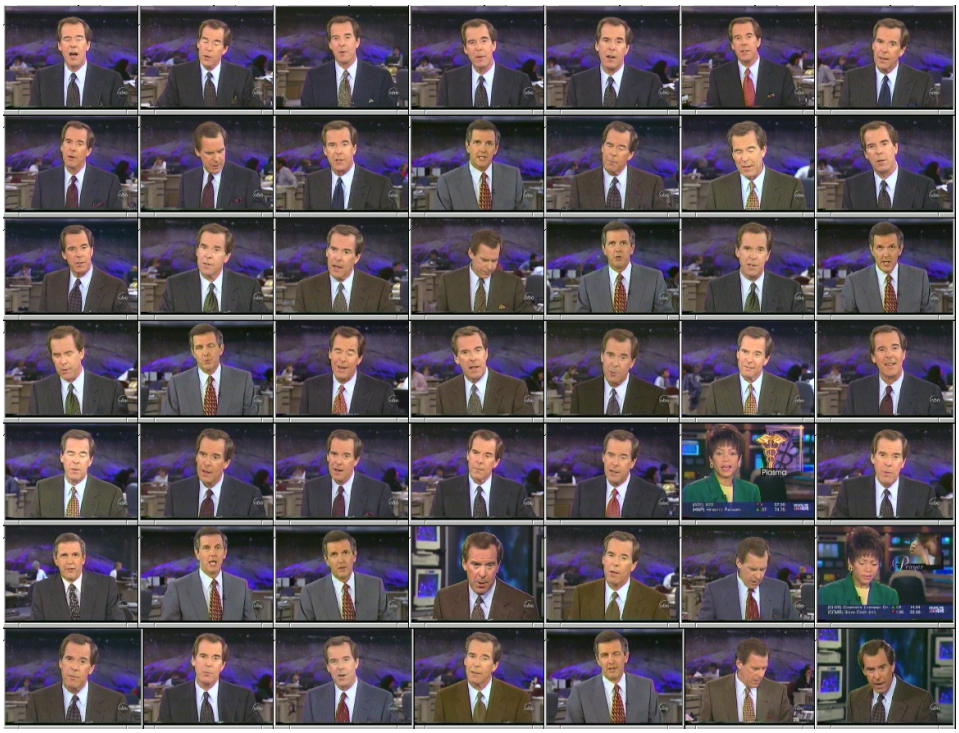

Figure 3.7: The top 49 shots returned by the anchorperson detector, clearly demonstrating its effectiveness and accuracy.

	Text Threshold	Anchor Threshold	Recall	Precision
shot-1	-	-	0.899	0.122
text-1	0.05	-	0.392	0.323
text-2	0.10	-	0.488	0.266
limsi-1	0.30	-	0.192	0.108
limsi-2	0.50	-	0.421	0.116
synonyms-1	0.05	-	0.308	0.324
synonyms-2	0.10	-	0.421	0.286
anchor-1	-	0.30	0.854	0.118
anchor-2	-	0.35	0.464	0.356
combined-1	0.05	0.30	0.375	0.319
combined-2	0.05	0.35	0.271	0.449

Table 3.7: Story boundary detection task — recall and precision results for the simple text-based system using the subtitle transcripts.

chains, as used in summarisation (see Chapter 6).

The results from the anchorperson-based system, tested in runs *anchor-1* and *anchor-2*, are promising, with precision being much higher than for text-based merging at corresponding recall levels. Note that these results do not necessarily reflect the accuracy or otherwise of the anchorperson detector. The original assumptions that story boundaries always lie on shot boundaries, and that every story begins with an anchorperson shot are not watertight.

System	Rec	Prec
DCU03_REQ_AV	0.352	0.427
DCU03_REQ_AV_TEXT	0.318	0.424
DCU03_REQ_TEXT_ONLY	0.048	0.186
DCU03_OPT_AV	0.337	0.426
DCU03_OPT_CLUSTER	0.386	0.298
Fudan_Story_Sys01	0.502	0.585
Fudan_Story_Sys02	0.502	0.585
Fudan_Story_Sys03	0.388	0.749
Fudan_Story_Sys04	0.388	0.749
Fudan_Story_Sys05	0.434	0.638
Fudan_Story_Sys06	0.434	0.638
Fudan_Story_Sys07	0.583	0.281
Fudan_Story_Sys08	0.288	0.307
Fudan_Story_Sys09	0.388	0.749
Fudan_Story_Sys10	0.277	0.779

System	Rec	Prec
IBM_CU_av_filter	0.568	0.770
IBM_CU_avt_filter	0.580	0.765
IBM_CU_t_filter	0.628	0.580
IBM_CU_av	0.531	0.735
IBM_CU_avt	0.552	0.774
IBM_CU_v_classification_only	0.211	0.362
kddi_ex1_10_10_1_n40	0.224	0.240
kddi_ex2_10_10_1_n40	0.217	0.214
kddi_noex_n40	0.234	0.235
NUS_1	0.727	0.736
NUS_2	0.745	0.739
NUS_3	0.720	0.720
NUS_4	0.735	0.752
NUS_5	0.491	0.545
ssudc1	0.231	0.285
ssudc2	0.249	0.244
ssudc3	0.048	0.186
UCFVISION	0.092	0.322
UIowaSS0301	0.287	0.696
UIowaSS0302	0.422	0.320
UIowaSS0303	0.253	0.224
UIowaSS0304	0.287	0.588
UIowaSS0305	0.482	0.305
UIowaSS0306	0.346	0.244
UIowaSS0307	0.366	0.381
UIowaSS0308	0.771	0.141

Table 3.8: Story boundary detection task — recall and precision results for all TRECVID participants.

The last two runs, *combined-1* and *combined-2*, are for the combined text and anchorperson system. A high and a low value for the anchorperson threshold were selected, along with a constant value for the text threshold that was seen to be optimum in testing on the development set. As expected, precision was increased — and the run *combined-2* returned the highest precision result (0.449) of any of our runs. In the future we hope to combine the two algorithms in a more sophisticated way, exploiting the strengths of both. In the current system, if the anchorperson detector causes a segment merge there is no way of "getting it back" at the next stage. It could be more effective to perform a weighted combination of evidence, rather than the sequential approach currently employed.

Anecdotally, the text based story detection is highly accurate in our live system that

captures and formats the BBC 10pm news daily. The high performance in this system may be, in part, as a result of the way in which the BBC subtitles are transmitted, and this is described in Section 6.2.1.

Table 3.8 shows the results for all runs submitted by the official TRECVID participants. Our unofficial runs compare well with most of the groups, though our simple approach is outclassed by the sophisticated techniques used by NUS [11] and IBM [2].

While our story boundary detection approaches are promising, there is much potential for future work. Lack of time meant that the textual approach and the anchorperson detection approach were combined in a very rudimentary fashion. Combination in a more sophisticated way — for example using a Bayesian network — could make for much more impressive results. Some time also needs to be spent analysing the output of the processes and finding out the reasons for failure. The most successful approaches used in TRECVID were modular and combined a great deal of domain specific information in order to produce their results. There is also a great deal of potential for improvement of the text based approach. We have mentioned the use of a gazetteer for expanding a segment's vocabulary of geographic terms. Improvement to the level of sophistication of the WordNet vocabulary expansion would also help, as would the deployment of lexical chain analysis rather than the simple word matching techniques. Several groups in TRECVID carried out successful boundary detection by looking for simple cue phrases that helped to delineate the boundaries.

Chapter 4

Retrieval by feature extraction from key frames

4.1 Introduction

The use of textual information derived from subtitles or speech recognition transcripts has been shown to be very effective in some domains, particularly broadcast news. However, transcripts only describe the audio part of the broadcast, and any information transmitted solely in the visual component is lost unless it is manually annotated. In this chapter we focus on how we address this problem by automatically deriving information from the *visual* component of the video.

There are a number of advantages to being able to use the visual content as the basis for a search:

- **A picture is worth a thousand words.** It is often difficult to fully express a visual query in words, and yet a single image can completely describe what is being searched for. Describing an object such as an aeroplane simply in terms of its shapes and colours would be a demanding task, yet providing an example can give all the information that is required. Similarly, one can easily visualise a tropical beach with palm trees and a clear ocean, but describing it clearly to a retrieval system without the use of examples would be a challenge. This notion does, of course, rely on having an example image to start with, so is only suitable for queries of the type "Find me more images like this one.".

- **Additional information.** Systems which employ subtitles or speech recognition are relying solely on the audio component of a broadcast, and while this provides a great deal of the necessary information in some domains, there is still much information that can only be gleaned from the visual component. A news report from London, with the Houses of Parliament in full view, will probably not have the words "Houses of Parliament" actually spoken, so anyone searching for a video clip containing the

Houses of Parliament will have no hook into this information. Many modern feature films rely far more on their visual effects than they do on any spoken material and a search based solely on the audio content could be very frustrating.

- **Language independence.** The use of visual cues allows the retrieval system to become language independent; an advantage where the database is to be made available internationally, such as on the internet.

4.2 System overview

A major problem with content-based retrieval of video material is the sheer volume of information which must be processed — at 25 frames per second a half hour broadcast contains 45,000 frames — and we address this problem by segmenting the video into shots, using the shot boundary detection procedure described in Chapter 3 and processing a single key frame to represent that shot. The set of key frames for a video broadcast can then be treated as a database of still images.

For each key frame we generate a number of feature vectors, as outlined in Section 4.3. When similar feature vectors have been generated for some query images, comparison of the query vectors with vectors for the database images forms the basis for retrieval. We have tested three retrieval algorithms on these feature vectors: i) a vector space model, ii) a nearest neighbour approach, and iii) a variation of the AdaBoost algorithm. The boosting method was initially chosen because it had shown promising results for still image retrieval [93] and we decided to compare this to the vector space model since this is an established method for vector comparison. We added the k-nearest neighbours approach since this had shown promising results in other work that we had carried out. Each of these methods is described in more detail in Section 4.4. Having retrieved a key frame associated to a video shot, the video can then be played by the user.

We have carried out two evaluations of these methods: an initial study using a database of 658 images divided into 31 categories and a more extensive study using a carefully constructed collection of over 6,000 still images divided into 63 categories.

The evaluations of the retrieval methods consisted of a number of retrieval and classification tasks, described in Section 4.5. The results of the evaluation are given in Section 4.6.

The initial study was presented at the International Conference on Image and Video Retrieval [71] and the more extensive later work was published in the Journal of Computer Vision and Image Understanding [69].

Information from the evaluation on the still-image data sets was used in the design of a system for the TREC 2002 Video Track and the TRECVID 2003 workshop. Our system and results from these workshops are presented in Section 4.7, further demonstrating the effectiveness of our approach.

While the key frame approach has proved to be very effective for representing video, there are some situations where it is not adequate, since the temporal aspects of camera and

object motion are lost, as well as the audio — i.e. all of the aspects which distinguish video from still images. Some of the TRECVID queries in which actions were expressed could not reliably be answered with a simple key frame approach — for example, "Find shots of a locomotive approaching the viewer." or "Find shots of an airplane taking off." Editors searching stock footage may well be searching for particular camera motions — for example, zooming in on an object of interest, or panning across a certain scene.

The key frame approach largely represents the state of the art in video retrieval at the current time, with one or two notable exceptions such as the Temporal Colour Correlogram devised by Rautiainen and Doermann [75], in which the spatio-temporal relationship of colours is calculated throughout a video shot. In our own approach, we make use of the audio track through use of subtitles and Automatic Speech Recognition, in combination with the still key frames.

4.3 Feature generation

Describing an object to a retrieval system typically involves the use of such characteristics as texture, colour and edge orientation. The vectors that we generate from the key frames are designed to encapsulate some or all of these in a global representation of the image. Each of the following sections describes a feature that we use in our experiments: convolution filters, HSV colour histograms, HMMD colour histograms and the Colour Structure Descriptor.

4.3.1 Convolution filters

The convolution filter feature is based on a set of 25 primitive filters described by Tieu and Viola [93], which are designed to respond to horizontal, vertical and diagonal edges and, at successive levels of filtering, different arrangements of these edges.

The process is shown diagrammatically in Figure 4.1. Each of the filters is applied to each of the three (RGB) colour channels of the key frame to generate 75 feature maps. Each of these feature maps is rectified and down-sampled before being fed again to each of the 25 filters to give 1875 feature maps. The process is repeated a third time, and then each feature map is summed, resulting in 46,875 feature values. The idea behind the three stage process is that each level 'discovers' arrangements of features in the previous level. The feature generation process is computationally quite costly, but only needs to be done once and then the feature values can be stored with the image in the database.

This process results in the definition of highly selective features which are determined by the structure of the image, as well as capturing information about colour, texture and edges. By defining a vast set of features, each feature is such that it will only have a high value for a small proportion of images, and by discovering a number of features which best characterise the example(s) we are able to perform an effective search.

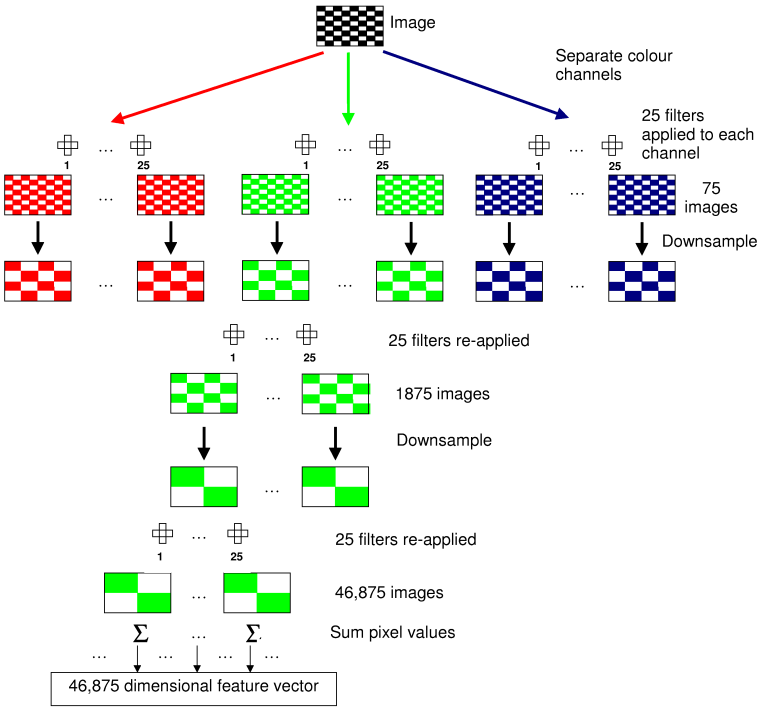

Figure 4.1: Generating the convolution feature vectors

4.3.2 RGB, HSV, HSL, Y'CbCr, CIELUV and CIELAB colour histograms

Retrieval from image databases using only colour was one of the first content-based retrieval methods — see for example Swain and Ballard [90], Faloutsos et al [24] and Pentland et al [66]. There is an abundance of colour spaces [42, 97, 109], virtually all of which are three-dimensional owing to the human perception of light using three different cones as receptors in the retina. Colour histograms are quantised distributions in the 3D colour space of all pixels of one image. The corresponding feature vector is a list of the proportions of pixels which fall into the respective 3D colour bins; its length depends on the granularity of the colour bins. We do *not* use one-dimensional component-wise histograms since (as with all marginalisations) information about the underlying colours would be lost.

RGB is a colour space in which colour is defined by adding certain proportions of red, green and blue light (see Figure 4.2); it is device dependent. Most images in digital image libraries are stored in some kind of RGB format in their native form and will have already incorporated some sort of gamma-correction (an adjustment made to compensate for the fact the light intensity produced in computer monitors is not proportional to the input voltage). The RGB colour space is a cube, all colours of which can be presented on a monitor.

Most other colour spaces can be derived from RGB by certain transformations. Y'CbCr is a linear transformation from gamma-corrected RGB values. Y' is a brightness component whereas Cb and Cr code the chromaticity of colour.

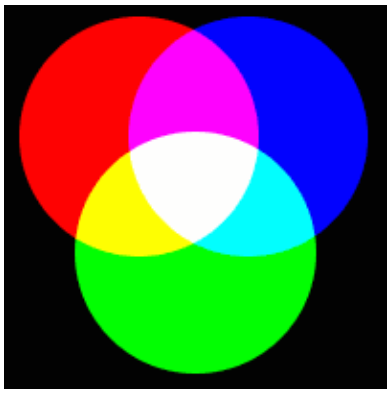

Figure 4.2: RGB is an additive colour space. (Source: wikipedia.org)

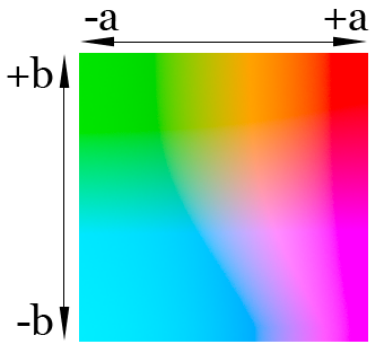

Figure 4.3: Variation of a and b components in the CIELAB at 75% luminance. (Source: wikipedia.org)

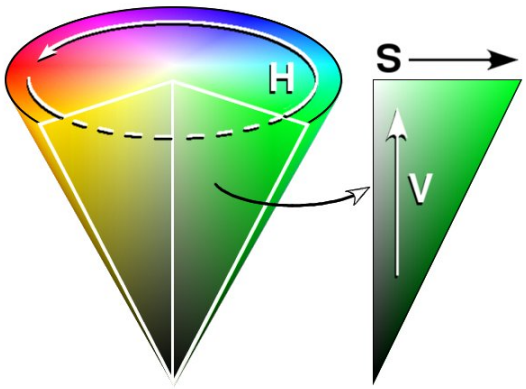

Figure 4.4: The HSV cone. (Source: wikipedia.org)

CIELUV and CIELAB [41] describe the physical appearance of colour, independent of the reproduction device, using the idea of a human standard observer. CIELAB is represented in Figure 4.3. Both colour spaces strive to be perceptually uniform in the sense that the Euclidean distance between any two points in colour space should resemble the human perception of the grade of colour difference. As a consequence, the transformations from RGB to CIELAB and CIELUV are non-linear and the resulting colour spaces are not of a simple geometric form. Hence, the same number of subdivisions of the CIELUV components will result in fewer "active" bins as compared to the same number of subdivisions on the RGB axes.

HSV and HSL [97] seem to be the most intuitive colour spaces to humans (HSV is shown diagrammatically in Figure 4.4). The hue coordinate H encodes the underlying pure colour tone of a colour circle. The saturation S reflects the pureness of the colour (the less pure the colour the more grey is mixed into it, S is zero for greys). V and L are both measures, albeit differently defined, for the apparent brightness or luminosity. When expressing the difference between two colours, humans tend to use HSL or HSV coordinates ("purer green", "darker crimson"), rather than RGB components.

HSV and HSL are both cylindrical colour spaces with H being the angular, S the radial and V or L the height component. This brings about the mathematical disadvantage that hue is discontinuous wrt RGB coordinates and that hue is singular at the achromatic axis $r = g = b$ or $s = 0$. As a consequence we merge, for each brightness subdivision separately, all pie-shaped 3D HSV bins which contain or border $s = 0$. The merged cylindrical bins around the achromatic axis describe the grey values which appear in a colour image and take care of the hue singularity at $s = 0$. Saturation is essentially singular at the black point in the HSV model and at both black and white points in the HSL model. Hence, a small RGB ball around black should be mapped into the bin corresponding to $hsv = hsl = (0, 0, 0)$, or $hsl = (0, 0, 1)$ respectively for white, to avoid jumps in the saturation from 0 to its maximum

m.a.p.	Colour space	Subdivisions
0.0955	HSV lin	$9 \times 9 \times 9$
0.0945	HSL lin	$9 \times 9 \times 9$
0.0944	HSV lin	$8 \times 8 \times 8$
0.0936	HSL lin	$8 \times 8 \times 8$
0.0933	RGB	$9 \times 9 \times 9$
0.0927	RGB	$8 \times 8 \times 8$
0.0894	CIELAB	$9 \times 9 \times 9$
0.0889	HSV vol	$9 \times 9 \times 9$
0.0883	CIELAB	$8 \times 8 \times 8$
0.0882	HSV vol	$8 \times 8 \times 8$
0.0880	CIELUV	$8 \times 8 \times 8$
0.0880	HSL vol	$9 \times 9 \times 9$
0.0868	HSL vol	$8 \times 8 \times 8$
0.0866	CIELUV	$9 \times 9 \times 9$
0.0844	Y'CbCr	$9 \times 9 \times 9$
0.0747	Y'CbCr	$8 \times 8 \times 8$

Table 4.1: Performance of simple colour histogram retrieval methods

of 1 when varying the singular RGB point infinitesimally.

There are several possibilities for a natural subdivision of the hue, saturation and brightness axes: they can be subdivided i) linearly, ii) so that the geometric volumes are constant in the cylinder and iii) so that the volumes of the nonlinear transformed RGB colour space are nearly constant. The latter refers to the property that few RGB pixels map onto a small dark V band but many more to a bright V interval of the same size; this is sometimes called the HSV cone in the literature. We have termed a subdivision according to i) HSV lin or HSL lin, and one according to iii) HSV vol or HSL vol respectively.

Precursory study on the choice of colour space

We carried out some experiments with category queries (see below) using mean average precision as a measure. We varied the number n^3 of bins in each colour space from $3 \times 3 \times 3$ to $10 \times 10 \times 10$ (but note that some colour spaces have effectively up to 50% fewer effective bins owing to their geometry). We found that, consistently across all colour spaces used, the respective performance began to flatten out at $n = 8$ or slightly peaked at $n = 9$. Table 4.1 shows the results for $n = 8$ and $n = 9$.

These experiments were done by averaging over 2517 different queries and 103 categories (using a different data set to the one used for our retrieval experiments described later, thus avoiding any bias towards the test data) — random retrieval corresponds to m.a.p. of 0.0108. We concluded that there is a consistent significant difference in the performance of

pure colour models, whereby the HSV/HSL spaces with a linear subdivision perform best and the perceptually uniform colour spaces, together with Y'CbCr, perform worst.

Based on this, we decided to settle for $8 \times 8 \times 8$ HSV colour space histograms with a linear subdivision of the axes. It has the smallest effective feature vector in the top 6 ranked methods with 456 components (less than $8^3 = 512$ owing to the merging of bins near the achromatic axis as mentioned earlier).

4.3.3 HMMD colour histogram

The new HMMD (hue, min, max, $diff$) colour space, which is supported by MPEG-7, is readily derived from the HSV and RGB spaces. The hue component is the same as in the HSV space, and max and min denote the maximum and minimum among the R, G, and B values, respectively. The $diff$ component is defined as the difference between max and min. Three components are sufficient to uniquely locate a point in the colour space and thus the space is effectively three-dimensional. Following the MPEG-7 standard, we quantise the HMMD space non-uniformly into 184 bins with the three dimensions being hue, sum and $diff$ (sum being defined as $(max + min)/2$). Manjunath and Ohm [52] give more details about quantisation. Two descriptors are defined with respect to the HMMD colour space. The first is a standard global histogram; the second, CSD or colour structure descriptor, is described in more detail below.

4.3.4 CSD: Colour Structure Descriptor

This descriptor lends itself well to capturing local colour structure in an image. An 8×8 structuring window is used to slide over the image. Each of the 184 bins of the HMMD histogram contains the number of window positions for which there is at least one pixel falling into the bin under consideration. This descriptor is capable of discriminating between images that have the same global colour distribution but different local colour structures. Although the number of samples in the 8×8 structuring window is kept constant (64), the spatial extent of the window differs depending on the size of the image. Hence, in line with MPEG-7, appropriate sub-sampling is employed for higher resolution images to keep the total number of samples per image roughly constant. The bin values are normalised by dividing by the number of locations of the structuring window and fall in the range $[0.0, 1.0]$ (Manjunath and Ohm [52] give details).

4.4 Retrieval methods

4.4.1 Vector space model (VSM)

Traditionally the vector space model [79] uses the entire feature vector, and images in the database are ranked according to the cosine of the angle between their feature vector and the sum of the feature vectors of the positive examples.

We use a variant of the vector space model in which the distance of two images q and t is defined as the Manhattan difference $\text{dist}_1(q,t)$ of the respective feature vectors. The distance of a query consisting of n images $q_1, \ldots, q_n$ to a test-set image t is defined as the sum of the respective $\text{dist}_1(q_i,t)$ distances.

Mathematically, all distances induced by the Minkowski norm L_n, $n \geq 1$, are equivalent (in finite-dimensional spaces) in the sense that they result in the same sort of topology for neighbourhoods, they define continuity of functions in the same way. Given a probabilistic model for the noise distribution in image features, one can compute an optimal distance function for retrieval tasks [83]. In the absence of a particular noise model, we note that amongst all L_n norms L_1 is the one which is most stable (i.e. least numerically influenced) against outliers and noise. Indeed we found that L_1 consistently outperformed L_2, L_∞ and the cosine similarity measure, which is nearly equivalent to the Euclidean distance for small distances.

4.4.2 Boosting

The AdaBoost algorithm [28] aims to select the most relevant features by determining which features generate the minimum error when classifying the example set.

We deploy Tieu and Viola's adaptation of this method [93] in which a hypothesis is determined from the positive and negative images by building a strong classifier from a weighted combination of weak classifiers. The ranking of an image in the database is based on how closely it fits the hypothesis.

A user specifies a query by selecting some positive examples. Negative examples are also required, but in a sufficiently disparate database, these can be randomly selected without too much danger of accidentally picking positive results. In the categorised image database used in our evaluation, we forcibly exclude from the negative set images which fall into the same category as the positive examples. Even when positive results are selected by chance, they can be removed in an interactive relevance feedback stage. The positive and negative example sets are then used to generate a strong classifier using the AdaBoost algorithm. The strong classifier is generated from the combination of a series of weak classifiers, each of which is determined by one iteration of the algorithm. An initial weight is assigned to each image, such that the total weights of the positive and negative images are equal, and weights are evenly distributed within these two groups. A hypothesis is generated at each iteration. Weights are re-assigned on successive iterations according to whether the image was correctly classified at the previous stage, so that more attention is paid to images which have thus far been incorrectly classified, and the hypotheses generated at each stage are combined to give an overall hypothesis which can be used to rank the images in the database. The complete algorithm, as used by Tieu and Viola and to which we have added our own hypothesis generation, is as follows:

- For example images (x_1, y_1),..., (x_n, y_n) where $y_i \in \{0,1\}$ for negative and positive images respectively, initialise weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively where m

is the number of positive images and l is the number of negative images.

- For $t = 1, ..., T$, where T is the number of weak learners upon which the final hypothesis is to depend:
  - Train a hypothesis h_j for each feature j using w_t, with error
    $$\epsilon_j = \sum_{h_j \neq y_i} w_{t,i}$$
  - The hypothesis h_j is determined as follows:
    * A threshold is calculated as the midpoint of the weighted mean of the values of feature j for the positive examples and the weighted mean of the values of feature j for the negative examples.
    * Images in the database for which the value of feature j falls on the same side of the threshold as the weighted positive mean or the weighted negative mean are classified by this weak learner as positive or negative, respectively.
  - Choose the hypothesis with the lowest error. Let $\epsilon_t = \epsilon_k$.
  - Update the weights:
    $$w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-e_i}$$
    where $e_i = 0$ for example x_i classified correctly and $e_i = 1$ for example x_i classified incorrectly. $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
  - Make w_{t+1} a distribution by normalising:
    $$w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{n} w_{t+1,j}}$$
- The final hypothesis is then
  $$h(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t$$
  where $\alpha_t = \log \frac{1}{\beta_t}$. $h(x)$ is used to rank the images.

4.4.3 K-nearest neighbours (k-NN)

We use a variant of the distance-weighted k-nearest neighbours approach [57]. As with the boosting method, a number of positive examples are supplied, and a number of negative examples are randomly selected from the database. The distances from the test image i to each of the k nearest[1] positive or negative examples are determined, and a relevance measure calculated as follows:

$$R(i) = \frac{\sum_{p \in P} (\text{dist}(i,p) + \varepsilon)^{-1}}{\sum_{n \in N} (\text{dist}(i,n) + \varepsilon)^{-1} + \varepsilon}$$

[1] 'nearest' is defined by the Manhattan (L_1) distance in feature space

where P and N are the sets of positive and negative examples respectively amongst the k nearest neighbours, such that $|P| + |N| = k$. ε is a small positive number to avoid division by zero. Images are ranked according to $R(i)$.

For the still-image evaluations, a value of $k = 40$ was used, based on preliminary empirical testing. In our later experiments for the TRECVID 2003 workshop (see Section 4.7) we set k to vary on a per-query basis to be equal to the number of example images supplied. At a small cost to accuracy, this had the advantage of significantly reducing the execution time for a search as it eliminated a costly sort operation.

4.5 Experimental set up

Since there exists no standard test bench for the evaluation of video retrieval methods, and making manual relevance judgements for an evaluation of any size would have been infeasible, given our resources, we used two collections of categorised still images to form the basis of evaluation.

4.5.1 Creating an image collection

Our initial still-image collection was created with the aim of being able to assess the performance of the retrieval methods on realistic search tasks. The initial collection was based on a small, little-known picture library and was sufficient for preliminary experiments. However, it is known that the choice of a test collection can heavily influence the perceived performance of a system [60], therefore it was considered important to use a large collection containing a wide range of images, and one which could easily be reproduced by other groups — hence a test collection was derived from one of the commercially available Corel products.

Initial study

Our initial study was performed using the Softkey PC Paintbrush photo library, which was reduced to 658 images, divided into 31 categories of conceptually (and often visually) similar images. Categories included planes, surfing, golf, bears, miscellaneous people, market stalls, horses, wolves, military and computers.

Corel collection

Our second set of experiments was designed to address some shortcomings with the original evaluation. Firstly, the test collection was too small to provide a realistic evaluation. Secondly, in the original evaluation, query images were drawn from the test set; this time around we partitioned the data so that query images were completely distinct from the test set. Finally, we wanted to perform our experiments on a widely used and widely available data set, both to give credibility to our results and to allow other groups to repeat the experiments for comparison. The Corel collection is already well established in the image retrieval community.

The improved collection was derived from the Corel Gallery 380,000 product, which contains some 30,000 photographs, sorted into around 450 categories. There were some problems with the existing Corel categorisation, and these had to be addressed in order to create a fair evaluation. A number of the categories were very similar to each other in content — for example, "Landscapes I" and "Landscapes II" — and in order to be able to assess the relevance of retrieved results it was necessary to remove this source of confusion. Some categorisations — for example, "Lifestyles" — were very abstract and contained such a diverse range of images that one could not realistically expect a query by example system to be able to retrieve effectively. With the aim of deriving a collection with significant *intra*-category similarity and *inter*-category difference, we reduced the collection to 63 categories, containing a total of 6,192 images by removing similar categories (favouring larger categories) and those with little or no degree of visual coherence within the category. This manual selection of categories helped ensure that there was no bias towards any particular automatic retrieval methods. The final list of category names that was used in our experiments is shown in Table 4.2.

The collection was then randomly split into 25% training data and 75% test data, once at the outset. This ensured that query images were outside the test collection and not counted amongst the results. Also, all training of the retrieval methods was done using training data only, if at all. All query image names, relevance judgements and category names used in our experiments can be downloaded from:

http://km.doc.ic.ac.uk/publications/image-evaluation-data/

and used by research groups who have acquired a copy of Corel Gallery 380,000.

As well as the convolution feature vector that was used in the initial study, we also carried out experiments using HSV colour histograms, the Colour Structure Descriptor, and HMMD colour histograms.

4.5.2 Experiments

We carried out a number of tasks in order to determine and compare the effectiveness of the use of different feature vectors and retrieval methods, reflecting the types of task that one might wish to perform in a video search.

In our initial study using the Picture Library data, we used category queries to assess retrieval effectiveness ("find me more examples of x").

In the extended study on the Corel data, we added two other tasks: classification tasks — assessing usefulness in annotation of video, and "real world" queries — queries constructed to determine the performance of the system on more specific information needs.

For the boosting and k-NN methods, 100 negative examples were randomly picked from the whole of the training set for each query, excluding the category to which the positive query image belonged. For k-NN, we used $k = 40$.

African Antelope	Museum China
Agates Crystals and Jaspers	Museum Dolls
Apes	Museum Duck Decoys
Beads	Museum Easter Eggs
Bears	Museum Furniture
Beautiful Roses	Mushrooms
Beverages	Nesting Birds
Bonsai and Penjing	New York New York
Bridges II	Orchids
Canadian Historic Railways	Ornamental Designs
Canadian Rockies	Painted Backgrounds
Cards	Pedigree Cats
Castles of Europe I	Pedigree Dogs
Caverns	Penguins
Clouds	Pill Backgrounds
Coastal Landscapes	Places of Worship
Contemporary Buildings	Plant Microscopy
Crystallography	Portraits
Doors of Paris	Prehistoric World
Fine Dining	Produce
Flora	Reflective Effects
Forests and Trees	Rhinos and Hippos
Insects	Roads and Highways
International Fireworks	Rome
Kitchens and Bathrooms	Sand and Pebble Textures
Lighthouses	Spectacular Waterfalls
Lions	Sunsets Around the World
Marine Life	Tools
Models	Volcanic Eruptions
Molecules	Wading Birds
Monument Valley	Zion National Park
Moths and Butterflies	

Table 4.2: The selected categories from the Corel collection, making up our test set

Category queries (Initial study and Corel study)

The category experiments were designed to determine how well the system is able to return images that are visually similar to the query. The judgement as to whether an image is similar to the query is based on whether it came from the same category.

Initial study. Firstly, each image in the database was used as a single-image query, and returned results were judged to be relevant if they belonged to the same category. For the boosting and k-NN methods, negative examples were randomly picked from the whole database, excluding the category to which the positive query image belonged.

For each category we then generated 6 random queries containing 2 positive examples from the category, and repeated the process for 3, 4 and 5 positive examples. For all queries the query images are removed from the returned list before calculating statistics.

In this initial study, only the convolution feature vector was used.

Corel study. For each of the retrieval methods, and for each type of feature vector, we used every one of the 1,548 images in the training set as a query and determined mean average precision for each method/vector combination. For each category, we also randomly generated 10 n-image queries from the training set for each n between 2 and 6 in order to determine the effect of using more example images. As there were 63 different categories, we generated 630 random n-image queries for each method/vector combination.

Classification tasks (Corel study only)

The Corel categories were grouped into nine general classifications, which are listed in Table 4.5, and 10 training sets were randomly generated for each classification, each containing 50 examples from the training data. Again, the judgement of relevance was based on whether or not a test image belonged to the same classification as the query. Note that the slightly ambiguous sounding classification "Objects" was largely a catch-all for remaining categories that did not fit another classification.

Real world queries (Corel study only)

The Corel collection comes with some manual annotation and a number of real world queries were constructed from key words in these annotations and, again, appropriate examples were randomly picked from the training set with which to query the database. 10 3-image queries were evaluated for each subject and mean average precision values obtained as before. A returned result was judged relevant if it also had the appropriate key words in its annotations.

Evaluation

A measure was required in order to evaluate the effectiveness of the system in these different tasks. We used average precision, which was described in Section 2.5.

Queries	Boost	VSM	k-NN	Random
1 image	0.1609	0.1946	0.1676	0.0536
2 images	0.1659	0.1343	0.1841	0.0408
3 images	0.1877	0.1397	0.2105	0.0387
4 images	0.1971	0.1247	0.2269	0.0379
5 images	0.2029	0.1299	0.2272	0.0361

Table 4.3: Mean average precision for retrieval for boosting, vector space model and k-NN, compared with a random ranking for the initial study using the convolution feature.

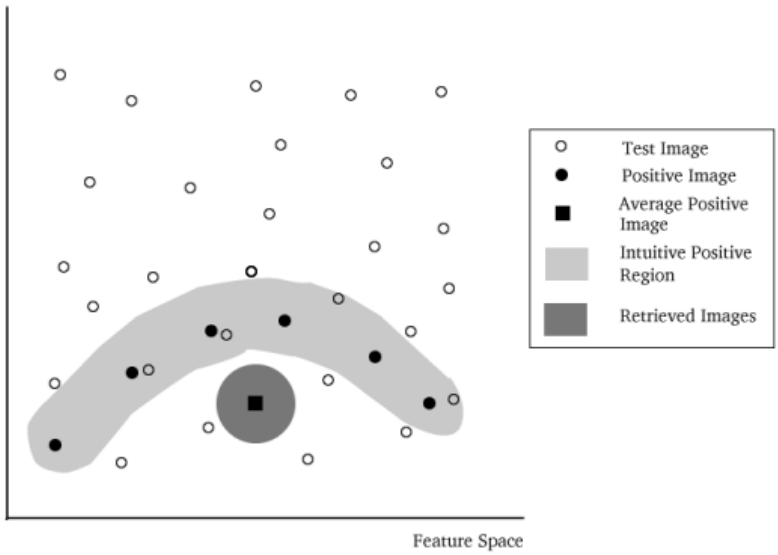

Figure 4.5: Illustration of degradation of VSM performance with multiple image queries

4.6 Results

4.6.1 Results from initial study

The results from the initial study are shown in Table 4.3. Paired t-tests performed on the data proved that all pairwise comparisons across the rows are statistically significant at an α level of 0.01, with the exception of the comparison between boosting and k-NN, for 1, 2 and 3 image queries.

The mean average precision values of all deployed methods are significantly better than random ranking of database images. The k-NN and boosting approaches returned particularly impressive results and, predictably, precision increased as more query images were used. The relative performance of the learning (boosting and k-NN) algorithms and the vector space model are interesting. The results show a clear trend of feature learning performing better with more image examples, while the vector space model deteriorates in performance. Figure 4.5 demonstrates the problem that we believe gives rise to these re-

sults. Since the positive feature vectors (represented by the solid black dots) are summed for the query in VSM, the resultant query (the black square) is effectively an average of the positive images. When the query images do not form a convex cluster the resultant query may be nowhere near any of the query images, so the returned images (represented by the dark grey shaded region) bear no relation to the images one would intuitively expect to have returned (represented by the lighter grey region). What this problem basically boils down to is that no mathematical representation can fully represent a human's view of similarity, nor encapsulate exactly which aspect of similarity the human wishes to search on. The mathematical average of a number of vectors does not necessarily accurately capture the human's view of what they intended to exemplify with that particular combination of images. k-NN intuitively ranks highly images found close to the positive query images, and this perhaps more accurately models what the human user is searching for — hence its superior performance for the multiple image queries.

4.6.2 Results from Corel study

Our results are presented in Tables 4.4 to 4.6, where mean average precision values are given for each of the experiments, using each different feature vector and retrieval method.

Paired t-tests performed on the data for the category and classification experiments proved that all pairwise comparisons across the rows are statistically significant at an α level of 0.05, for all differences in mean average precision greater than 0.02.

Category queries

The category query results are shown in Table 4.4. As with the initial study, the mean average precision values of all methods were significantly better than random ranking and precision increased as more query images were used. The performance of the three methods was largely similar, though k-NN outperformed boosting in all cases and the VSM in all cases except for the single-image queries. We believe this is due to the random selection of query images in categories which contain visually disparate images. Boosting works best when the query images are visually similar. As observed in the initial study, k-NN intuitively ranks highly images found close to the positive query images and so its performance is good if the query images are representative of the category being evaluated, even when visual similarity between them is not good. As the number of query images is increased in the VSM, there is no actual degradation in performance as there was in the initial study, but the increase in performance with more query images is less marked for the VSM than for the other two methods, again probably due to the averaging of multiple images to form the query.

As with most information retrieval tasks, the spread in performance over different queries was enormous. For some queries in categories with little or no visual connectedness the average precision could be as low as with random ranking. For many categories, however, where visual connectedness was good, the methods excelled and the average precision for some individual queries was as high as 98%.

The superior performance of the HSV feature vector was perhaps surprising, given the simplicity of its computation. This observation suggests that it would be worth investigating the creation of convolution features using the HSV space instead of RGB — especially as the convolution features were found to be superior in some situations outlined in the next section.

Classification tasks

The mean average precision results for the classification tasks are shown in Table 4.5. The relative performance of the three retrieval methods is much in line with that seen in the category queries, with k-NN providing the superior results. However there are some interesting comparisons to be made between the feature vectors. When the query is heavily colour based (for example, food, flowers, patterns) the colour based vectors (HSV in particular) outperform, or are comparable to, the convolution vector. However, when colour is a less important factor, and perhaps more information is deduced from the structure of the image (for example, architecture, landscape) the convolution vector outperforms the colour based vectors. Nevertheless, it is perhaps surprising to note that the colour based vectors do still give a respectable performance in classifications such as "Architecture". This may largely be due to the fact that some of the categories did contain very similar images, perhaps even of identical objects.

In general, the results here are much better than for the category queries, which may in part be due to the greater number of training images used, as well as to the fact that there are far fewer classifications than there were categories in the previous task. It is not unrealistic to expect that a large number of training images could be collected for classification tasks, since in order to build a classifier this task only has to be performed once.

Real world queries

The results for the three real world queries that we carried out on our test collection are shown in Table 4.6. Once again k-NN mostly outperforms boosting and the VSM. As with the classification tasks it is remarkable how different the performance of different features can be. This supports the hypothesis that a relevance feedback step in the search process could inform the search engine which *features* contribute dominantly to relevant images. Together with a large range of different, orthogonal features one would expect such a relevance feedback process to markedly increase the performance of *individual* queries, while our category analysis helps to build a system with increased performance on *average*.

The superior performance of k-NN relative to boosting in the category experiments is probably due to the way in which the queries are constructed. k-NN copes well with visually disparate positive examples since positive results can be found in disjoint clusters. The boosting algorithm depends on discovering visual similarity in order to build a good retrieval hypothesis.

Results from this evaluation have been used in the design of systems for the search

Queries	Vector	Boost	VSM	k-NN
1 image	Convolution	0.1043	0.1327	0.1548
	HSV	0.1195	0.1813	0.1959
	HMMD	0.1061	0.1348	0.1467
	CSD	0.0945	0.1382	0.1530
2 images	Convolution	0.1376	0.1664	0.2003
	HSV	0.1586	0.2142	0.2360
	HMMD	0.1280	0.1466	0.1631
	CSD	0.1108	0.1535	0.1767
3 images	Convolution	0.1606	0.1736	0.2262
	HSV	0.1807	0.2286	0.2543
	HMMD	0.1442	0.1555	0.1790
	CSD	0.1264	0.1613	0.1943
4 images	Convolution	0.1759	0.1825	0.2474
	HSV	0.1978	0.2345	0.2717
	HMMD	0.1516	0.1604	0.1940
	CSD	0.1420	0.1644	0.2072
5 images	Convolution	0.1819	0.1889	0.2526
	HSV	0.2171	0.2436	0.2801
	HMMD	0.1619	0.1581	0.1979
	CSD	0.1487	0.1658	0.2120
6 images	Convolution	0.1858	0.1943	0.2659
	HSV	0.2285	0.2484	0.2924
	HMMD	0.1669	0.1635	0.2109
	CSD	0.1558	0.1707	0.2220

Table 4.4: Mean average precision for the category tasks for boosting, vector space model and k-NN for each of the feature vectors, using the Corel data set. The mean average precision for random ranking of images is 0.0175

Class	Vector	Boost	VSM	k-NN	Random
Animals	Convolution	0.3718	0.3753	0.5179	0.1925
	HSV	0.3992	0.3024	0.4862	
	HMMD	0.3465	0.2639	0.4423	
	CSD	0.3504	0.2570	0.4061	
Architecture	Convolution	0.3361	0.2742	0.4318	0.0906
	HSV	0.2500	0.2025	0.2897	
	HMMD	0.2827	0.1703	0.3028	
	CSD	0.2448	0.2167	0.2601	
Flowers	Convolution	0.1836	0.0963	0.3505	0.0700
	HSV	0.3544	0.1770	0.4154	
	HMMD	0.3283	0.0870	0.2998	
	CSD	0.3232	0.0742	0.3764	
Food	Convolution	0.0998	0.1533	0.2371	0.0517
	HSV	0.2848	0.3145	0.2586	
	HMMD	0.2948	0.2118	0.2272	
	CSD	0.2706	0.2621	0.2640	
Insects	Convolution	0.1251	0.1068	0.1543	0.0552
	HSV	0.2783	0.1483	0.4489	
	HMMD	0.3482	0.0903	0.3285	
	CSD	0.2944	0.1020	0.3917	
Landscape	Convolution	0.3152	0.2093	0.3536	0.1427
	HSV	0.2936	0.1554	0.2859	
	HMMD	0.2710	0.1599	0.2437	
	CSD	0.2491	0.1417	0.2354	
Objects	Convolution	0.4730	0.3071	0.5162	0.1709
	HSV	0.4511	0.4369	0.6289	
	HMMD	0.4723	0.2702	0.6230	
	CSD	0.4425	0.2367	0.5404	
Patterns	Convolution	0.3104	0.1478	0.2959	0.1132
	HSV	0.3771	0.2878	0.3800	
	HMMD	0.3871	0.1641	0.3453	
	CSD	0.4635	0.1717	0.3841	
People	Convolution	0.1204	0.1147	0.1337	0.0326
	HSV	0.1930	0.1553	0.1854	
	HMMD	0.1097	0.1154	0.0891	
	CSD	0.0630	0.0782	0.0974	

Table 4.5: Mean average precision for the classification tasks for boosting, vector space model and k-NN, compared with a random ranking, for each of the feature vectors.

Query	Vector	Boost	VSM	k-NN	Random
Sunset	Convolution	0.3499	0.2270	0.3419	0.0197
	HSV	0.1155	0.1088	0.1696	
	HMMD	0.1787	0.1140	0.1550	
	CSD	0.1097	0.1767	0.2662	
Lion	Convolution	0.0653	0.1575	0.1849	0.0156
	HSV	0.0866	0.1179	0.1267	
	HMMD	0.0489	0.0708	0.0863	
	CSD	0.0517	0.0702	0.1073	
Door	Convolution	0.4365	0.4591	0.5524	0.0179
	HSV	0.3340	0.5097	0.5377	
	HMMD	0.1256	0.2688	0.2583	
	CSD	0.0585	0.2087	0.2041	

Table 4.6: Mean average precision for the real world tasks for boosting, vector space model and k-NN, compared with a random ranking, for each of the feature vectors

tasks of the TREC 2002 video track [98] and the TRECVID 2003 workshop [99], which are described in Section 4.7. The results of the TREC evaluation further demonstrate the effectiveness of these methods on real video queries [68].

4.6.3 Limitations of this evaluation

Some major shortcomings of the initial evaluation were overcome with the extended evaluation using the Corel data, but although the test collection was constructed very carefully to give a good spread of images and reliable relevance judgements, some problems remain.

The Corel categorisations were not designed for this kind of experiment, and the categories tend to represent conceptual similarities rather than visual similarities. For example, pictures of the Eiffel Tower and the Louvre could be placed in a category called "Paris", but one could not expect a content-based retrieval system to group them together. While we attempted to remove categories which were likely to give rise to this kind of problem it is clear that many categories could still contain images that are uncharacteristic of the rest of the category.

The annotations upon which we based the relevance judgements for the "real world" task were also found to be difficult to work with; the annotations were neither complete nor consistent (for example, one image labelled "outdoors", another labelled "outside"). An "outdoors" image retrieved by our system which had not been labelled as such would be judged as a wrongly retrieved image.

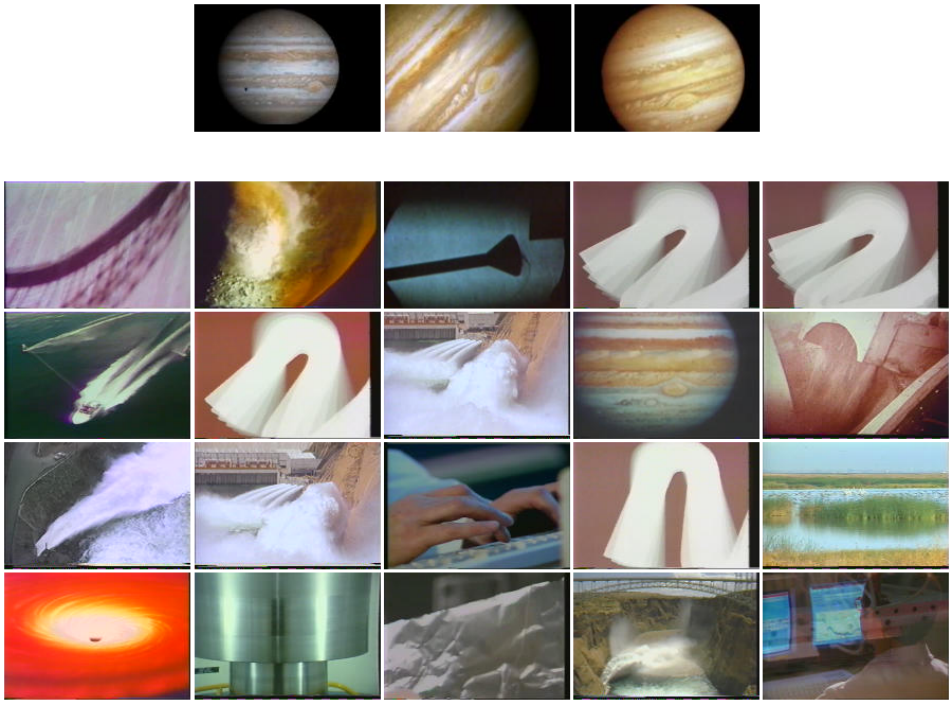

Figure 4.6: Query images and first 20 returned results for the Jupiter query, using the boosting method

4.7 Experiments at TREC and TRECVID

The TREC Video Track and TRECVID frameworks provided us with an ideal opportunity to perform an effective evaluation of our system and to compare it against other groups' systems.

In the search task, video search systems were presented with 25 topics (formatted descriptions of an information need) and required to return a ranked list of up to 1000 shots from the search test collection.

In the following sections, we outline our anecdotal experiments with a topic from the 2001 Video Track, and systems for the 2002 TREC Video Track and the TRECVID 2003 workshop, and present the results obtained in the evaluations.

4.7.1 TREC 2001

We did not participate in the search task of the 2001 Video Track, but tested the boosting method, using the convolution filter feature, on a topic in which the requirement was to find shots of the planet Jupiter.

Figure 4.6 shows the positive images supplied for the query and the results returned by our system, where the Jupiter known-item appears ninth.

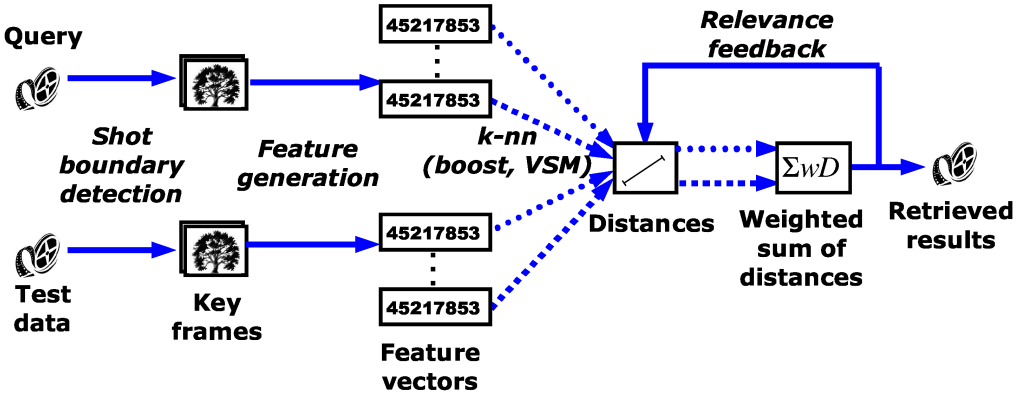

Figure 4.7: Retrieval by feature extraction — system overview

4.7.2 TREC 2002

Data

The 2002 data for the search task consisted of just over 40 hours of data from the Internet Archive [92]; mainly advertising, educational, industrial and amateur films from the 1930s-1970s.

System

Our system for the TREC 2002 Video Track search task was heavily influenced by our earlier experiments using the Corel database. The superior performance of the k-NN method in those experiments led us to adopt this as the underlying retrieval method, and to use combinations of the features with weights informed by the cross-feature comparisons. An overview of how we deploy our retrieval methods is shown in Figure 4.7.

To the four features used in the Corel evaluation, we added two others: an illumination-invariant feature designed by O'Callaghan and Bull [62], and a text feature based on the Automatic Speech Recognition transcript generated by LIMSI [29]. A full text index was built using the Managing Gigabytes search engine [58] and queries formed from the XML data supplied with each topic. Managing Gigabytes supplies a numerical relevance value which was used when weighting features.

Retrieval was carried out within a system designed by Heesch et al [35], in which video shots are represented by key frame thumbnails displayed on the screen. Distance from the centre of the screen is inversely proportional to the relevance of that shot. The user can provide relevance feedback by changing the position of the thumbnails on the screen before the system re-evaluates the ranking. More details of the relevance feedback process, devised by Daniel Heesch, can be found in our TREC proceedings paper [68].

Experiments

Topics were supplied with a mix of text description, example videos and example images. For the example videos, we extracted appropriate key frames using shot boundary detection, and added these to the example images.

We carried out four runs over the 25 topics (see Table 4.7) to investigate the effects of various combinations of features and of relevance feedback:

1. All features + using relevance feedback.
2. Illumination invariant, Text and Convolution features only.
3. All features. (Baseline for run 1).
4. CSD, Text and Convolution features only. (Baseline for run 2).

Topic no	Description
075	Find shots with Eddie Rickenbacker in them.
076	Find additional shots with James H. Chandler.
077	Find pictures of George Washington.
078	Find shots with a depiction of Abraham Lincoln.
079	Find shots of people spending leisure time at the beach, for example: walking, swimming, sunning, playing in the sand. Some part of the beach or buildings on it should be visible.
080	Find shots of one or more musicians: a man or woman playing a music instrument with instrumental music audible. Musician(s) and instrument(s) must be at least partly visible sometime during the shot.
081	Find shots of football players.
082	Find shots of one or more women standing in long dresses. Dress should be one piece and extend below knees. The entire dress from top to end of dress below knees should be visible at some point.
083	Find shots of the Golden Gate Bridge.
084	Find shots of Price Tower, designed by Frank Lloyd Wright and built in Bartlesville, Oklahoma.
085	Find shots containing Washington Square Park's arch in New York City. The entire arch should be visible at some point.
086	Find overhead views of cities — downtown and suburbs. The viewpoint should be higher than the highest building visible.
087	Find shots of oil fields, rigs, derricks, oil drilling/pumping equipment. Shots just of refineries are not desired.
088	Find shots with a map (sketch or graphic) of the continental US.
089	Find shots of a living butterfly.

Topic no	Description
090	Find more shots with one or more snow-covered mountain peaks or ridges. Some sky must be visible them behind.
091	Find shots with one or more parrots.
092	Find shots with one or more sailboats, sailing ships, clipper ships, or tall ships — with some sail(s) unfurled.
093	Find shots about live beef or dairy cattle, individual cows or bulls, herds of cattle.
094	Find more shots of one or more groups of people, a crowd, walking in an urban environment (for example with streets, traffic, and/or buildings).
095	Find shots of a nuclear explosion with a mushroom cloud.
096	Find additional shots with one or more US flags flapping.
097	Find more shots with microscopic views of living cells.
098	Find shots with a locomotive (and attached railroad cars if any) approaching the viewer.
099	Find shots of a rocket or missile taking off. Simulations are acceptable.

Table 4.7: TREC Video Track 2002 search task topic descriptions.

Results

In Table 4.8 we show the average precision results for our 4 runs across the 25 topics. The topics for which we achieved our best results are, at first glance, surprising — topics 75 and 76 were both queries requiring specific personalities, Eddie Rickenbacker and James H Chandler respectively. Our system was not designed to detect faces. However, both queries contained film of quite distinctive colouring, and the Chandler query contained query shots from within the test set.

Using our four runs we hoped to show that relevance feedback improved performance and that the use of illumination invariant features improved performance, but results were not completely conclusive for either hypothesis. In Run 1, the user was allowed to perform relevance feedback by moving images towards or away from the centre and then re-running the query as many times as necessary before settling on a final results set. This is compared to Run 3, in which the same features were used but the user had to settle for the first set of results returned, without relevance feedback. In Run 2 we used the illumination invariant feature, along with the convolution and text features only. In Run 4, for comparison, the illumination invariant feature was replaced with another colour feature, the Colour Structure Descriptor.

As we carried out our interactive (relevance feedback) run (Run 1) the retrieved shots were certainly visually much better with each round of relevance feedback, though this is not spectacularly clear from the numerical results. There was some improvement in the average precision for most topics, an observation which is reinforced by the 95% confidence interval for the difference between the performance means of the results for this run and its baseline

Topic	Run 1	Run 2	Run 3	Run 4
75	0.172	0.146	0.142	0.146
76	0.487	0.540	0.545	0.442
77	0.000	0.005	0.000	0.000
78	0.000	0.188	0.000	0.172
79	0.003	0.000	0.002	0.002
80	0.081	0.009	0.146	0.071
81	0.138	0.000	0.000	0.000
82	0.005	0.005	0.004	0.014
83	0.133	0.028	0.000	0.024
84	0.260	0.250	0.050	0.258
85	0.000	0.000	0.000	0.000
86	0.067	0.048	0.022	0.066
87	0.004	0.065	0.000	0.046
88	0.075	0.001	0.006	0.017
89	0.050	0.000	0.000	0.019
90	0.040	0.105	0.075	0.058
91	0.000	0.020	0.000	0.000
92	0.121	0.021	0.011	0.033
93	0.002	0.007	0.001	0.001
94	0.017	0.001	0.005	0.006
95	0.005	0.003	0.000	0.005
96	0.017	0.004	0.014	0.001
97	0.067	0.053	0.050	0.033
98	0.003	0.000	0.000	0.002
99	0.091	0.001	0.000	0.004

Table 4.8: TREC 2002 Video Track Search task — average precision results.

(Run 3) which, while not proving statistical significance, does suggest an improvement in performance using relevance feedback. The perceived performance improvement may simply be due to the fact that relevance feedback re-ordered the rankings — and with better top-ranked results the user's overall impression is one of greater satisfaction. Some topics (for example, 81 — football players, 83 — Golden Gate Bridge, 88 — US maps) benefitted significantly from the application of relevance feedback. The interactive runs in TREC model a search scenario where someone, such as a librarian, searches on behalf of someone else who ultimately judges the returned results. This is different from our model of relevance feedback where the searcher is the one who judges and uses the results. It is also important to note that a user may often be more content with one or two good results, highly ranked, than with retrieving every relevant item in the database.

Calculation of the 95% confidence interval for the difference between the means of the results of the illumination invariant run (Run 2) and its baseline (Run 4) showed that the introduction of the illumination invariant feature brought about no overall improvement in results. However, performance was improved in a number of specific topics, for example, topics 90 (snow covered mountains), and 91 (parrot).

With hindsight, the experiments could have been better designed; in some cases the limited number of features used in the second run performed better than the combination of all features (Run 3), suggesting that a philosophy of "more features is better" does not necessarily hold. Some further experiments could be carried out to discover which combinations of features work best — whether there are some features that are consistently good and some that are consistently unhelpful, and whether some features facilitate good results in the presence or absence of other particular features.

4.7.3 TRECVID 2003

Data

The 2003 test data consisted of 113 video files; a mix of CNN and ABC news, and C-SPAN documentary programmes. The video was partitioned into 32,318 shots and a key frame supplied for each. The video data was accompanied by an Automatic Speech Recognition transcript, supplied by LIMSI [29].

System

Our TRECVID 2003 search system built on our earlier work at TREC 2002, and particular attention was paid to the interactive part of the system, adding features to optimise our ranked list of results (so that the user could create a 'shopping list' of relevant shots, rather than just returning the last list that the system came up with) and adding browsing functionality [36]. The main interface, shown in Figure 4.8 was similar to the one employed in 2002, and again uses distance from the centre to represent relevance of returned shots. Two further browsing interfaces were added on top of this system to aid the search.

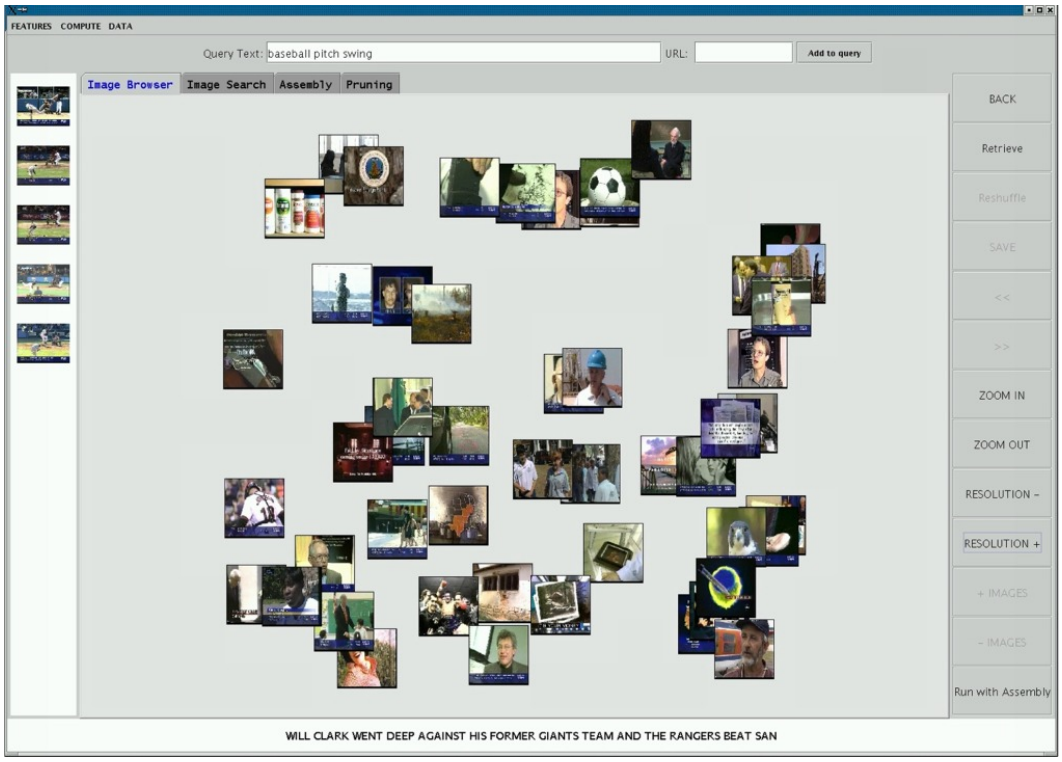

Figure 4.8: TRECVID 2003 search interface — distance from the centre is inversely proportional to relevance.

Figure 4.9: TRECVID 2003 system, temporal browser — when the user mouses over an image, this sliding window consists of the image and its left and right shot neighbours.

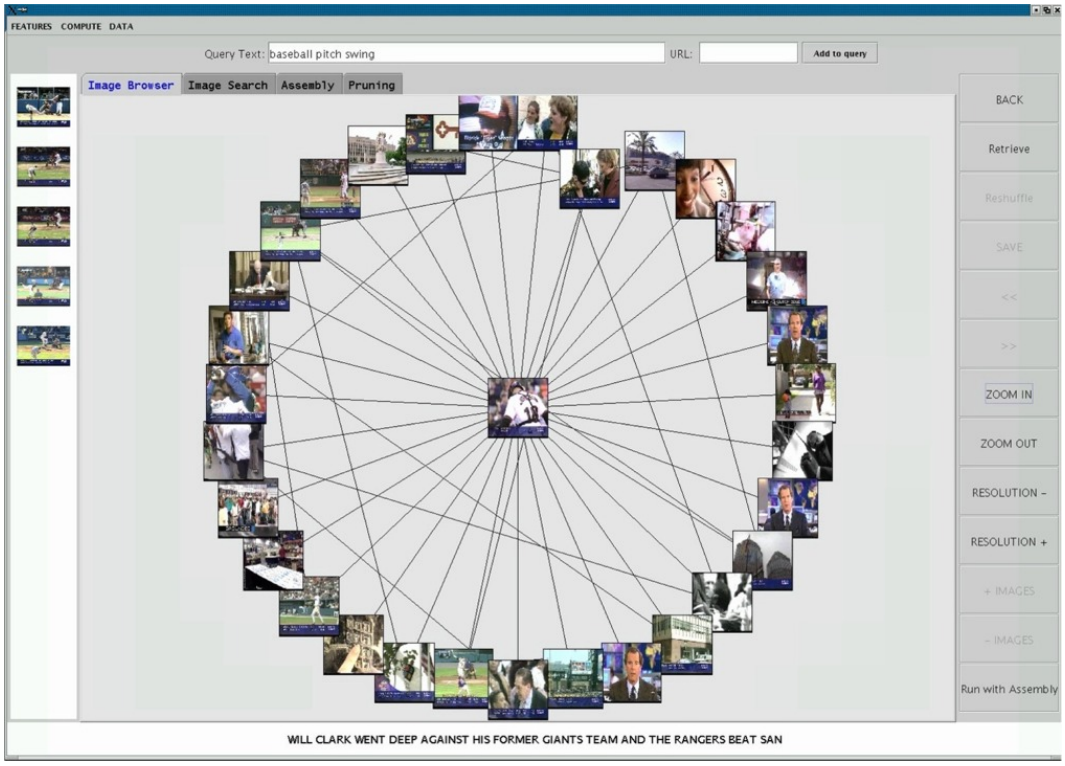

Figure 4.10: TRECVID 2003 system, lateral browsing interface — two images are connected if there exists at least one feature combination for which one image is ranked top when querying with the other.

The *temporal browser*, shown in Figure 4.9, enabled the user to move temporally backwards and forwards through the broadcast. The rationale behind this is twofold: firstly, shots of interest often occur clustered near to each other in a broadcast and, secondly, some retrieval methods — particularly where based on the transcriptions — do not always accurately pinpoint exactly the correct shot, but rather one that is very near to the relevant shot. The temporal browser allows the user to search in the neighbourhood of retrieved shots. It initially displays the current shot key frame with its left and right shot neighbours. Clicking the << and >> buttons allows the user to move through the broadcast.

The *lateral browser*, shown in Figure 4.10, allows the user to move through the shots based on per feature connections. Two images are connected by a line if there exists at least one feature combination for which one image is ranked top when querying with the other. This allows the user to browse the network intuitively using different feature combinations. Initially the top ranked image for each feature combination is displayed in a random position in the circle around the query image.

The features that we used were calculated on the shot key frames, each with the bottom 52 pixels removed. This is because much of the material had a ticker across the bottom of the screen displaying headlines or other material not relevant to the shot in question. We used the convolution, HSV colour histogram, HMMD and CSD features previously described, and

added a number of others:

HSV Focus Histogram. Like the described HSV histogram, but using only the central 25% of pixels from the key frame.

Marginal RGB Colour Moments. Individual histograms were formed for each of the RGB colour channels and the mean and second, third and fourth central moments computed for each marginal colour distribution.

Thumbnail. Obtained by scaling down each key frame and recording the grey value of each pixel. This feature is particularly suited to identifying near-identical key frames — e.g. when they appear in commercials etc.

Variance. A 20 bin histogram of grey value standard deviations was computed in a 5×5 sliding window for each of 9 image tiles.

Smoothness. A smoothness measure was calculated for each of 64 image tiles.

Uniformity. A uniformity measure was calculated for each of 64 image tiles.

Bag of words. Using the textual annotation obtained from the LIMSI transcripts, we computed a bag-of-words feature consisting for each image of the set of accompanying stemmed words (Porter's algorithm) and their weight. This weight was determined using the standard tf-idf formula and normalised so that the sum of all weights was 1.

Experiments

As in 2002, topics were supplied with a mix of text description, example videos and example images. For the example videos, key frames were extracted as before.

Topic no	Description
100	Find shots with aerial views containing both one or more buildings and one or more roads.
101	Find shots of a basket being made — the basketball passes down through the hoop and net.
102	Find shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at.
103	Find shots of Yasser Arafat.
104	Find shots of an airplane taking off.
105	Find shots of a helicopter in flight or on the ground.
106	Find shot of the Tomb of the Unknown Soldier at Arlington National Cemetery.
107	Find shot of a rocket or missile taking off. Simulations are acceptable.
108	Find shots of the Mercedes logo (star).
109	Find shots of one or more tanks.

Topic no	Description
110	Find shots of a person diving into some water.
111	Find shots with a locomotive (and attached railroad cars if any) approaching the viewer.
112	Find shots showing flames.
113	Find more shots with one or more snow-covered mountain peaks or ridges. Some sky must be visible behind them.
114	Find shots of Osama bin Laden.
115	Find shots of one or more roads with lots of vehicles.
116	Find shots of the Sphinx.
117	Find shots of one or more groups of people, a crowd, walking in an urban environment (for example with streets, traffic and/or buildings).
118	Find shots of Congressman Mark Souder.
119	Find shots of Morgan Freeman.
120	Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day. The number of points risen that day must be visible.
121	Find shots of a mug or cup of coffee.
122	Find shots of one or more cats. At least part of both ears, both eyes, and the mouth must be visible. The body can be in any position.
123	Find shots of Pope John Paul II.
124	Find shots of the front of the White House in the daytime with the fountain running.

Table 4.9: TRECVID 2003 search task topic descriptions.

Two manual runs were carried out across the 25 topics (see Table 4.9). One was a compulsory run, with only the use of text permitted. For the other run we used topic-specific weight vectors to weigh each of the eleven features described above. The weight vectors were our crude guesses of the optimal weight sets and were based on inspection of the query images only.

Four interactive runs were carried out which differed with regard to the type of user interaction allowed. Temporal browsing formed a core functionality that was enabled in all runs. Users were allowed a maximum of 15 minutes for each topic in the interactive runs, though for several of the topics much less time than this was actually used. Four users each performed the queries for six different topics from each of the four runs.

1. Browsing: The users were not allowed to formulate any query but were allowed to see the query images and the text annotation that came with each topic. Images were selected by employing the browsing structure only.

2. Search + Browsing + Relevance Feedback: The users had available the full functionality of the system. They could freely formulate a query by selecting features, adding

	mean average precision	rank out of 38
TRECVID Best	0.218 ± 0.168	
TRECVID Median	0.0720	
TRECVID Mean	0.0851 ± 0.0665	
T	0.074 ± 0.1125	19
T + V	0.076 ± 0.0937	18

Table 4.10: TRECVID 2003 Search task — results for the manual runs averaged over all 25 topics (ranks based on mean average precision). T= text, V = visual features.

	mean average precision	rank out of 36
TRECVID Best	0.4573 ± 0.276	
TRECVID Median	0.1939	
TRECVID Mean	0.182 ± 0.088	
S + R + B	0.257 ± 0.219	5
S + R	0.259 ± 0.210	4
S + B	0.234 ± 0.240	8
B	0.132 ± 0.187	27

Table 4.11: TRECVID 2003 Search task — results for the interactive runs averaged over all 24 topics (ranks based on mean average precision). S = search, R = relevance feedback, B = lateral browsing.

or removing query images and by modifying the text query. Users could attempt to optimise the weight set by performing relevance feedback on retrieved objects. In addition the users had full access to the browsing structure.

3. Search + Browsing: Same functionality as above but without relevance feedback.

4. Search + Relevance Feedback: Browsing was not allowed, but users could search and optimise weights using relevance feedback.

Results

The 2003 topics are shown in Table 4.9, a summary of our manual results in Table 4.10, and a summary of our interactive results in Table 4.11. A detailed breakdown of our performance in all topics is shown in Table 4.12, where our average precision results are compared to the median performance of all systems. Relative to other systems, our performance was vastly improved over the previous year's results, particularly in the interactive task — where 3 of the runs were in the top 8 of all systems.

Particularly impressive results were returned in topics 100, 102 and 117, where our S+R system variant achieved the highest performance of all systems. Predictably, these topics

Topic	Interactive					Manual		
	Medn	B	S+R+B	S+B	S+R	Medn	T	T+V
100	0.030	0.117	0.090	0.086	0.142	0.008	0.000	0.005
101	0.064	0.039	0.225	0.033	0.359	0.041	0.002	0.002
102	0.131	0.429	0.398	0.211	0.513	0.111	0.010	0.059
103	0.287	0.000	0.586	0.443	0.474	0.122	0.167	0.127
104	0.091	0.127	0.115	0.058	0.132	0.029	0.059	0.051
105	0.175	0.043	0.347	0.162	0.239	0.024	0.249	0.019
106	0.316	0.595	0.447	0.791	0.326	0.154	0.123	0.251
107	0.197	0.286	0.345	0.242	0.194	0.037	0.017	0.096
108	0.187	0.001	0.302	0.321	0.132	0.063	0.091	0.052
109	0.125	0.001	0.250	0.077	0.125	0.005	0.152	0.152
110	0.077	0.186	0.226	0.161	0.090	0.002	0.030	0.003
111	0.151	0.000	0.005	0.186	0.001	0.002	0.002	0.001
112	0.120	0.030	0.116	0.063	0.156	0.029	0.020	0.036
113	0.145	0.149	0.316	0.335	0.144	0.022	0.006	0.008
114	0.346	0.001	0.472	0.516	0.477	0.119	0.118	0.119
115	0.089	0.006	0.142	0.103	0.158	0.014	0.006	0.014
116	0.707	0.585	0.839	0.850	0.882	0.122	0.511	0.083
117	0.021	0.057	0.059	0.020	0.081	0.020	0.023	0.037
118	0.000	0.000	0.000	0.000	0.347	0.000	0.000	0.000
119	0.000	0.000	0.000	0.000	0.000	0.013	0.000	0.000
120*	-	-	-	-	-	0.230	0.090	0.292
121	0.054	0.002	0.057	0.076	0.220	0.002	0.012	0.012
122	0.036	0.248	0.228	0.232	0.272	0.017	0.014	0.012
123	0.255	0.000	0.212	0.274	0.477	0.114	0.120	0.154
124	0.500	0.403	0.658	0.606	0.544	0.051	0.030	0.32

Table 4.12: TRECVID 2003 Video Track Search task — average precision results per topic. *Topic 120 was not included in the test set for interactive runs.

were requirements for a class of global scenes — buildings, crowds etc, rather than specific entities. Interestingly, the S+B system variant was the highest performer in topic 106, where the requirement was more specific; clearly the browsing functionality facilitated a narrowing of the search.

Searches for people were hard to execute. In the 2002 task the characteristic colours in the people searches gave us somewhat unexpected success in those topics — but this time around the quality of the video ensured that no such tactics could be employed!

When one or more relevant shots had been found, the lateral browsing structure made it easy to gather more relevant shots — particularly where colour and texture were important (e.g. topics 100 and 102), and the temporal browser meant that exact shots of interest could be pinpointed by searching shot neighbours.

Although our search results in TRECVID 2003 were impressive, and comparable to all other participating groups, our global approach has limitations. For some queries, where a specific object, person or type of motion is being searched for, we could really have benefitted from the deployment of more specialist modules. In particular, more use could be made in future of the features donated by other groups, where we would simply need to work on how to combine this evidence into our system. In the broadcast news domain there is much to be gained from being able to extract the text captions that are embedded in the pictures as this is a useful way of identifying personalities in the absence of (or in support of) face recognition technology.

Chapter 5

Colour classifiers

5.1 Introduction

This work originated from a 10 week summer internship at AT&T Laboratories, Cambridge UK, as part of the Personal Media Management (PERMM) project. The aim was to design improved colour classifiers for the Ontological Query Language (Oquel) developed by Town and Sinclair [96].

The work was split into two areas: a scheme for labelling regions with colour names, and a system for classifying regions into one of a number of visual categories, such as grass, sky, skin etc. Work had already been done on the classification using neural networks [95]. However, it was typically difficult to trace the source of misclassifications and it was felt necessary to redesign the classifiers with a greater degree of predictability at times of failure.

The segmentation scheme described in the next section was developed by David Sinclair; all other work described in this chapter is my own, and was carried out under supervision at AT&T.

5.2 Colour labelling

In order to facilitate effective retrieval, it is necessary to segment an image into logical regions and a scheme developed by Sinclair [84] was employed, which isolates regions according to colour, texture and edge information and an example of its operation is shown in Figure 5.1. The aim was then to associate a colour name with each region to provide indexing terms for keyword retrieval.

The HSV space was chosen for a combination of its simple mapping from RGB space, and its close modelling of human intuition about colour. It is straightforward to make adjustments to a query like "more red" and "brighter". Colour labelling was then carried out by defining a palette of 36 colours, and mapping each image region to the name of its closest colour in the palette, defined by the Euclidean distance in HSV space.

Figure 5.1: The image on the left has been segmented and regions identified as shown on the right

5.3 Classification into visual categories

5.3.1 Designing the classifiers

To facilitate more detailed queries, it becomes necessary to give higher level classifications to image regions. A set of 11 visual categories were chosen which were thought to represent a wide range of well-defined types of 'stuff'. Thee 11 categories were: grass, cloudy sky, blue sky, wood, sand, skin, trees, water, snow, brick and tarmac.

Regions were first clustered in RGB space to encapsulate the colour distribution into a few points. At first a fuzzy clustering algorithm [5] was used but proved to be very slow, and this was quickly replaced by a sequential clustering algorithm, in which a point was added to an existing cluster if it fell within a certain distance of the centroid (and then the centroid updated), or a new cluster created, with that point as the centroid, otherwise.

For each class, a set of a few hundred regions which positively identified that class, and a few hundred negative regions, were gathered and clustered. This training set was pruned by removing all those distributions for which all the nearest neighbours were of the same classification (positive or negative). This helped to ensure that as far as possible the distributions that were kept were the ones around the decision boundary.

A test region could then be classified by clustering it, and finding its 25 nearest neighbours in the training set. The test region was then labelled with the classification of the majority of its nearest neighbours.

The term 'nearest' is defined by the *earth mover's distance* [78], and is the cost of morphing one distribution into another. Figure 5.2 shows a 2-dimensional example. The circles

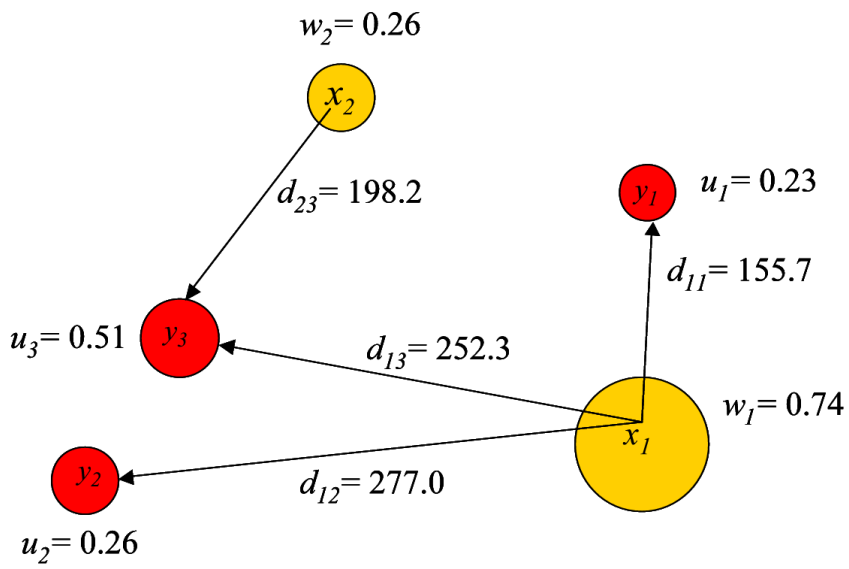

Figure 5.2: 2-dimensional example of the calculation of the Earth Mover's Distance

represent clusters, which are weighted by the proportion of the distribution's population belonging to that cluster. Weight is moved across the shortest possible distances first, so 0.23 is moved from x_1 to y_1 first, then 0.26 from x_2 to y_3 (being the next shortest distance), and so on until all the weight is moved. The total cost is the sum of the products of the weight moved and the distance it was moved. The nearest neighbours can be found by finding the smallest earth mover's distances. A region can be assigned a probability of membership of any of the visual classes by the proportion of the 25 nearest neighbours which match the classification.

5.3.2 Performance

Subjectively the classifiers worked well, although formal evaluation has thus far only been carried out for the 'grass' classifier.

Anecdotal tests showed that the 'grass' classifier was particularly effective, correctly labelling grass and non-grass regions in many images. Figure 5.3 shows the golf example, and the regions detected as grass and as water. However, since the classification is done solely based on colour, there are a number of predictable reasons for failure. Things which are of a similar colour to the stuff being searched for are often falsely positively labelled — for example in Figure 5.3, the 'grass' classifier (centre picture) detects some of the trees as 'possibly' grass and the 'water' classifier (right hand picture) detects darker parts of the bank as 'possibly' water. There are other notable areas of failure with other classifiers: Wood and skin are surprisingly similar in shade. Haze and distance affect the colours in photographs — many things take on a blue tint in aerial photographs, for example. Another problem is

Figure 5.3: Example of the operation of the 'grass' (centre) and 'water' (right) classifiers. Regions shown as green are those detected as grass/water with some certainty. Yellow and red regions are those which fall close to the decision boundary — yellow falling on the positive side and red on the negative side. All other regions are definitely not grass/water, according to the classifiers.

that some of the visual classes do not have a well-defined colour; tarmac, for example, can be anything from pitch black to light grey, depending on age, and was even seen to have a blue shade due to lighting conditions in some photographs. Bright materials like snow and water tend to reflect the colours of their surroundings, or are heavily affected by shadow and other changes in lighting conditions.

The classifiers were designed to err on the generous side, such that they were likely to produce false positives, with the aim that downstream classifiers could further refine the decision.

TRECVID

The TRECVID workshop included a "feature extraction" task, with the aim of detecting various semantic concepts in video material. Amongst the detection tasks for TRECVID 2003 was the feature "vegetation". Since our classifiers are based solely on colour, it was thought that the grass classifier might work just as well on general vegetation as it does on grass.

We submitted two runs, one using voting-based k-NN — in which a region's relevance score was based on the proportion of the nearest neighbours that are positive, and the other using distance-weighted k-NN — in which the distance of the nearest neighbours from the test region was also factored in.

Figure 5.4: The colour-based nature of the classifier means that false positives are generated when non-vegetation green areas appear.

For voting-based k-NN, a region's relevance score is calculated as:

$$R(i) = \frac{|P|}{k}$$

where P and N are the sets of positive and negative nearest neighbours respectively, such that $|P| + |N| = k$, and k is the number of nearest neighbours.

For distance-weighted k-NN, the relevance score is calculated as:

$$R(i) = \frac{\sum_{n \in N} (\text{dist}(i,n) + \varepsilon)^{-1}}{\sum_{p \in P} (\text{dist}(i,p) + \varepsilon)^{-1} + \varepsilon}$$

where $\text{dist}(i,p)$ and $\text{dist}(i,n)$ are the earth mover's distances from the test region to the positive and negative nearest neighbours respectively, P and N are the sets of positive and negative nearest neighbours respectively, such that $|P| + |N| = k$, and k is the number of nearest neighbours.

A binary decision on the existence of vegetation had to be given on a per *shot* basis, rather than a per *region* basis as we had been doing before. Therefore we processed the key frame for each shot, and calculated a relevance score which was the square of the highest region score in the key frame. The results were sorted by score, and top the 2000 submitted as positive shots, respecting the TRECVID requirement that a maximum of 2000 shots be submitted. k was set to 15 in these experiments.

The results are shown in Table 5.1. Examples of the types of images retrieved are shown in Figures 5.4 and 5.5. Because the classifier is based solely on colour distribution in regions, false positives were likely to occur — for example the computer screen shown in Figure 5.4. However, recall was good and lots of vegetation regions were retrieved, see for example Figure 5.5. Although voting-based k-NN had been shown empirically to give better results on the

Figure 5.5: Recall is good, for example detection of the grass area at this baseball game

System Variant	Avg prec	Hits
Imperial-01 (voting)	0.082	342
Imperial-02 (distance-weighted)	0.087	360
TRECVID Median	0.150	367

Table 5.1: TRECVID 2003 High level feature extraction task — average precision results and hits at depth 2000 for feature 16 (vegetation)

development data, it suffered in this case from the limit of 2000 submitted results. Over 2000 shots were judged by 100% of the nearest neighbours to contain vegetation and there was no possibility for ranking within this results set, so the only option was to truncate the results list at 2000. Although distance-weighted k-NN had given poorer results in testing, the distances gave a greater spread of scores and hence a less arbitrary ranking. The weakness of our ranking is reflected in the fact that our number of hits at depth 2000 was very close to the median achieved in all TRECVID runs, but our average precision fell well below the TRECVID median.

5.3.3 Future work

Before TRECVID 2003 there was a collaborative effort amongst participating groups to manually annotate the development data for training purposes [49]. In the future we should exploit this resource in order to better train our system. The vegetation classifier is likely to gain significantly from being trained properly with vegetation examples (rather than grass), and with so much training data to hand it should be possible to attempt some of the other features.

The results of the colour classifiers could be fed to other classifiers working on characteristics such as shape and texture in order to improve their precision.

Classifiers such as these could in the future replace much of the manual annotation effort which takes place for large image and video databases, facilitating straightforward keyword searches. This would circumvent the need for an initial query image, as is the case for query-by-example systems, and could allow a user to formulate natural language queries.

Chapter 6

News summarisation and retrieval systems

6.1 Introduction

The research that we have done in the areas of video segmentation and retrieval using features in key frames has been showcased with additional work in two live systems for retrieval and summarisation of news video.

The Automatic News Summarisation and Extraction System (ANSES) provides web-based retrieval and browsing of broadcast news, mainly using text methods for retrieval. The system utilises our shot and story boundary detection work to present the daily news on a story by story basis. Shot level keyframes provide a visual summary of the content of each story, and we have combined this with existing text summarisation techniques to facilitate intuitive and simple search and browsing of television news online.

The Aetos system, developed by Heesch et al [34, 36] builds on our earlier work in feature-based image retrieval to provide content-based image search, and we demonstrate the potential for this system to sit alongside the ANSES system to allow even further flexibility in search and retrieval of broadcast news video.

6.2 ANSES system

ANSES aims to present news video in such a way that relevant stories can be retrieved and browsed easily. Video and subtitles (closed-captions) are captured for the BBC news each night and the video is split into its constituent shots. The text is summarised, and information extracted from it is used in the merging of shots to form stories. The data capture and processing stages are summarised in Figure 6.1 and detailed in Section 6.2.1. Retrieval takes place through a web-based interface and is described in Section 6.2.2.

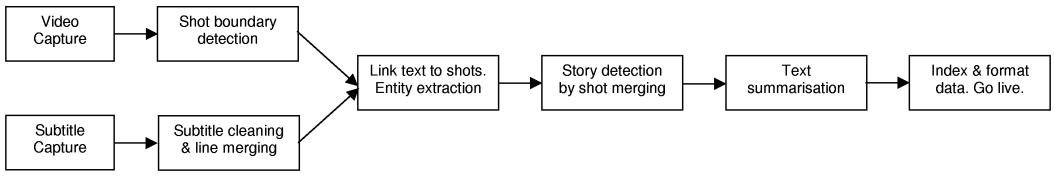

Figure 6.1: ANSES data capture and processing overview.

```
91.913088 An attack
92.912983 An attack on the UK is
93.912868 An attack on the UK is now, it
94.912876 An attack on the UK is now, it seems,
95.995791 An attack on the UK is now, it seems, inevitable. That
```

Figure 6.2: Subtitle grows word by word in subsequent lines.

6.2.1 Data capture and processing

Video and subtitle capture

The BBC 10pm news is automatically captured every night, along with the associated subtitles. The capture system is controlled by a cron script on a Linux-based PC containing a Hauppauge WinTV PCI card. Video recording is performed using the *streamer* application from the xawtv suite [44], and subtitle capture using a modified version of the *AleVT* software [94]. Once the capture of the raw video and unformatted subtitles has completed, the processing stages are then automatically started.

Video shot boundary detection

Video shot boundary detection is carried out by the process described in Section 3.2. A single key frame is generated for each shot.

Subtitle cleaning and line merging

Poor quality subtitles or speech recognition transcripts often contain sufficient information to provide the necessary keywords to facilitate acceptable information retrieval performance. However, the language processing techniques that we employ rely on having good quality

```
60.433061 Good evening. Britain's most senior policeman,
60.995228 Britain's most senior policeman, Sir John
62.073082 Britain's most senior policeman, Sir John Stevens, has spelled out
62.953060 Sir John Stevens, has spelled out the nature of
63.953063 Sir John Stevens, has spelled out the nature of the terrorist threat.
66.392871 the nature of the terrorist threat. The Metropolitan Police
```

Figure 6.3: Line is a partial duplicate of the previous line.

```
108.313087 Counter-terrorist officers on the strets of London
109.112912 Counter-terrorist officers on the streets of London today, are we
```

Figure 6.4: Growing line, but with mismatch in duplicated part.

```
80.912854 London's Mayor, Ken Livingstone, speaking at the same
81.912859 speaking at the same news conference
83.153141 nes conference said it would be "miraculous"
```

Figure 6.5: Partial duplicate, but with mismatch in duplicated part.

text in complete sentences. Some processing of the captured subtitle text was necessary to render it in a format suitable for entity detection and summary generation.

As well as imperfections caused by interference in the broadcast, a number of problems were also caused by the way that live subtitles are transmitted, with many duplicate phrases and lines (which appear seamlessly when displayed on a TV screen). These problems are illustrated in Figures 6.2 — 6.5. The first two figures show normal, error-free, transmission of subtitles: Figure 6.2 shows the case where one or more words are added to each transmitted line, and Figure 6.3 additionally shows lines that partially overlap with each other — for example, the phrase "Britain's most senior policeman," appears in the second part of the first line, and again as the start of the second line. It would be reasonably simple to match these line fragments if it were not for the addition of a further problem, shown in Figures 6.4 and 6.5: Here again we have the growing and overlapping subtitle lines respectively, but this time there is the additional complication that (probably due to transmission errors) there is a mismatch between the duplicated sections of text.

In order to solve this problem, Marc Lehmann's Perl String::Similarity module was employed to detect lines which were (possibly error-containing) duplicates, or partial duplicates, of preceeding and succeeding lines. The similarity function returns a value of 1 if two strings are identical and 0 if they are entirely different, with all other values in between based on the edit distance between the two strings.

In order to find out whether two lines have any overlap and should be merged, an initial comparison is carried out to determine the possibility of a match. Lines which display match potential are compared in detail, as shown in Figure 6.6, to find a merge point if one exists. The two lines are zipped across each other, and the window where the two strings intersect is examined using String::Similarity. With each iteration the window size (which starts with one word) is increased by one word. The iteration with the highest match score is used to indicate the merge point, providing the score exceeds a threshold (we use 0.8). If the threshold is not exceeded on any iteration, then no merge is made between these two subtitle lines.

In Figure 6.6 the iteration on which a match occurs is shown in italics. To form the new line, we take the string to the left of the window as the start of the line. If match score in the

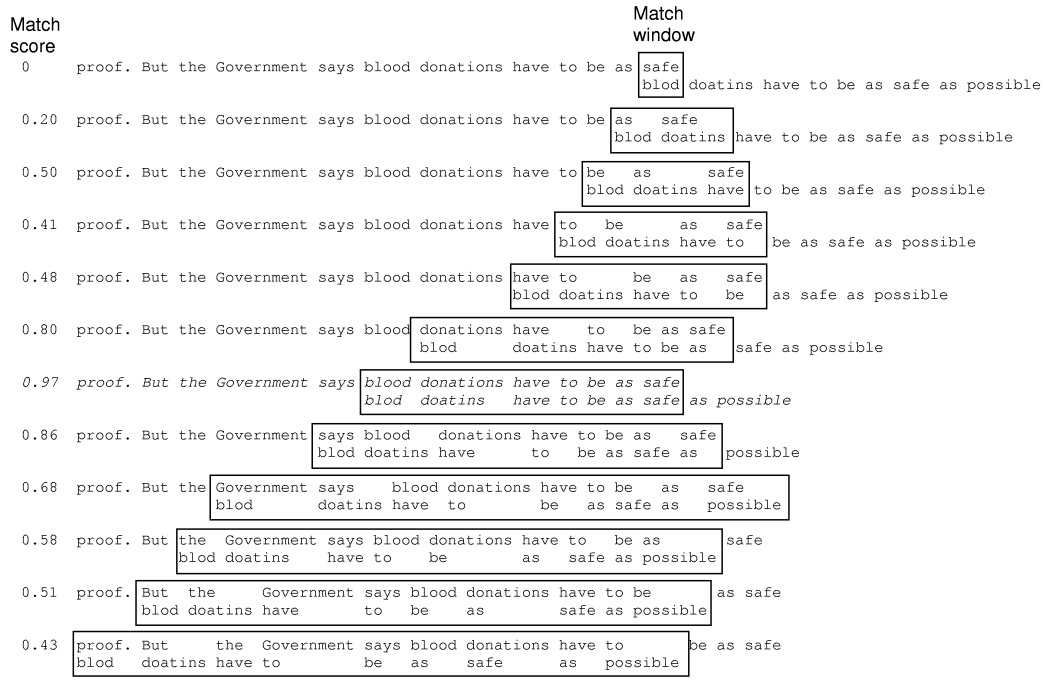

Figure 6.6: Partial line matching algorithm

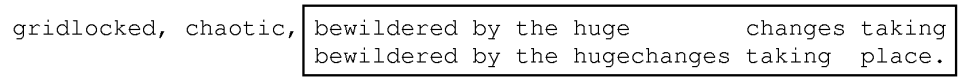

Figure 6.7: Partial line matching algorithm has problems when errors cause word merges

window is not 1 (i.e. an identical match) we take the string with the *most characters*, since subtitle errors are usually characterised by omission of characters (in the example, "blood donations" has become "blod doatins" through the omission of characters. The correct string is the longer one). This choice does fall down where the omitted character is a *space* as words will then have been merged, leaving an uneven match in the merge window — this problem is shown in Figure 6.7. The merge window size is based on the number of words and the system wrongly counts the merged "hugechanges" as one word. A merge will occur at the correct place at the left of the window, but the wrong string will be chosen from the merge window — giving rise to the output string "gridlocked, chaotic, bewildered by the hugechanges taking place. place.". Fortunately this is a relatively rare occurrence.

Following the subtitle cleaning process, subtitle lines are matched, according to their timestamps, with the shots delineated by the shot boundary detection process.

In live news broadcasts, the subtitles often lag several seconds behind the video and audio, and further work needs to be carried out in future to properly re-align the subtitles.

Figure 6.8: Key entities are highlighted alongside each story summary

This has previously been done by aligning the subtitles with the output of a speech recogniser [106]. However, a useful side-effect of the problem of overlapping subtitles, described above, is that overlap usually only occurs *within a story*, and the merging of lines results in a natural grouping of sentences within stories.

Key entity detection

The process of extracting key entities from the text associated with the video serves two purposes. Firstly, the key entities can provide a simple and consistent means for summarising the content of a story. Secondly, they provide information that is useful in determining which shots to merge to form stories.

Key entity extraction is performed using the General Architecture for Text Engineering (GATE) [19]. Keywords are extracted from the subtitles and tagged with their parts of speech (noun, verb, pronoun etc) and entities classified, where appropriate, as organisations, dates, people and locations.

Key entities are highlighted for stories in the interface as shown in Figure 6.8.

Story boundary detection

The process of story boundary detection is described in detail and evaluated in Section 3.3.

Text summarisation using lexical chains

Following the process of story boundary detection, each segment should contain a complete news story, and we wish to provide an accurate summary of the news story, so that the user

can glean the important content without reading it in its entirety. The intent here is not to develop a new text summarisation method, but rather to show how text summarisation can be used to summarise video, and so we implement an established summarisation technique using lexical chains [59]. The algorithm we have devised is inspired by Barzilay and Elhadad [4]. In general there are three stages for constructing lexical chains:

1. Selecting a set of candidate words.
2. Finding an appropriate chain for each candidate word, depending on a *relatedness* criterion among members of the chain.
3. Placing words. If an appropriate chain is found, the word is inserted in the chain, which is updated accordingly. If no appropriate chain is found, a new one is created, consisting only of this candidate word.

The *relatedness* of two candidate words is determined by the distance between their occurrences and the shape of the path connecting them in the WordNet thesaurus [55]. There are three types of relations: *extra strong* — between a word and its repetition, *strong* — between two words connected by a WordNet synonymy or hyponymy relation, and *medium strong* — where the link between synsets of the words is longer than one. When deciding in which chain a candidate word should be inserted, extra strong relations are preferred to strong relations, and both are preferred to medium strong relations. For each chain, a *chain score* is defined as the sum of the scores generated by each link in the chain. The score for a link is generated according to its type.

In our implementation, all nouns in the story are selected as candidate words. Nouns are detected using GATE's Part of Speech tagger,for which 80-85% precision and recall are claimed for news text. Each time a new noun is considered, we look up all the meanings of that noun. A different set of chains (an interpretation) must be considered for each meaning of each noun. On average, a noun has 8 different meanings, so for a typical story containing 30 nouns, there are $8^{30} = 1.238 \times 10^{27}$ interpretations. This represents a huge search space and necessitates some pruning. After every noun is added, the list of interpretations is sorted in descending order of the interpretation scores, where the interpretation score is the sum of the chain scores for that interpretation. The top 20 interpretations are then kept, the rest discarded.

Once we have considered all nouns in the story, we choose the interpretation with the highest score to represent the story. From this interpretation, we select the 3 strongest (highest scoring) chains, and from each of these chains we choose the word with the highest occurrence as a representative word. For every sentence that this word appears in a score is calculated using the following function:

$$\text{sentence score} = \sum_i n_i w(i),$$

where n_i is the number of occurrences of key entity i in the sentence and $w(i)$ is the weight associated with the type of word i (shown in Table 3.6). The two highest scoring sentences are used to represent the chain.

For each text segment we also prepend the *first* sentence of the full text of that segment, since the first sentence often gives a good summary on its own. This sentence is shown in italics in our interface before the automatically generated summary for each clip (see Section 6.2.2.

Data formatting and web preparation

The final stage of the data processing is to prepare the data for dissemination on the internet. The video and subtitle text are converted into RealMedia and RealText formats respectively and a SMIL [104] description file generated. This format was chosen as SMIL presentations can be played using the RealPlayer software, which is available for most platforms. The full text index for the text search engine is regenerated in order to incorporate the latest subtitle text, and the new day's news goes live on the web. All of the information about a broadcast is also stored in an XML file, consolidating the data and enabling future indexing of the data in a digital library. A truncated example is shown in Figure 6.9.

6.2.2 Retrieval

The retrieval interface is web based, and users have the choice of two retrieval methods — a date-based view and a keyword-based view.

Date-based view

In the date-based view, shown in Figure 6.10, the news is displayed on a day by day basis, with the stories ordered as they were broadcast, which gives a similar feel to that of browsing an online newspaper.

The default front page of the system displays the most recent day's news. Since the system currently works on the BBC 10pm news, the most recent news is generally from the night before. A simple date selector allows the user to move to any other day stored in the database quickly and intuitively.

Keyword-based view

In the keyword-based view, shown in Figure 6.11, a user can search using query words or phrases as one would do in a search engine such as Google.

As well as entering a keyword or phrase, the user has the option to limit the date range of the returned results (allowing a search, say, of the past week's news, or a specific known time period). It is also possible to specify whether the returned results are sorted by date (most recent first) or according to the relevance judged by the Managing Gigabytes search engine [58], which provides the back-end to the keyword search based on a full text index of the subtitle transcripts. Search terms are highlighted in red, both in the summary and in the lists of key entities.

```
<programme>
<programme_title>BBC 10 o'clock news Tuesday 13/1/2004 22:00</programme_title>
<date_time>Tuesday 13/1/2004 22:00</date_time>
.
.
<story start="57.04" end="572.52">
<story_title>Good evening.</story_title>
<story_keyframe>130104/130104/130104_0005.jpg</story_keyframe>
<smil_file>130104/130104/130104_0005.smil</smil_file>
<ram_file>130104/130104/130104_0005.ram</smil_file>
<realvideo>130104/130104/130104_0005.rm</realvideo>
<rawtext>130104/130104/130104_0005.vst</rawtext>
<summary>
Good evening. Two investigations are under way tonight into the death of
<person>Harold <person>Shipman</person></person>, <location>Britain</location>'s
most prolific serial killer. <person>Dr Harold Shipman</person> was jailed for
life after killing hundreds of his patients n injecting them with diamorphine
and pretending they died naturallally, though in the police interviews he denied
the murders. After conviction, Dr Harold Shipman was on suicide watch and was
checked every 15 minutes, but since he moved to Wakefield Prison, he was checked
hourly, which is about normal for a category A prisoner. Harold Shipman worked
as a family GP for more than 20 years in Hyde, <location>Greater
Manchester</location>. <person>Suzanne Brock</person> shows me the memories of
the grandmother she loved and that <person>Dr Shipman</person> killed.
</summary>
<full_text>
Good evening. Two investigations are under way tonight into the death of
<person>Harold <person>Shipman</person></person>, <location>Britain</location>'s
most prolific serial killer. He was found hanging in his cell at Wakefield
Prison this morning. The former family doctor was convicted nearly four years
.
.
nothing like this wil happen again.
</full_text>
<shot start="57.04" end="88.92">
<keyframe>130104/130104/130104_0005.jpg</keyframe>
<shot_text>
Good evening.
Two investigations are under way tonight into the death of Harold Shipman,
Britain's most prolific serial killer.
He was found hanging in his cell at Wakefield Prison this morning.
The former family doctor was convicted nearly four years ago of murdering 15 of
murdering 15 of his elderly patients.
But it's believed he killed as many as 260 people.
He'd always denied his crimes.
In a moment, the reaction from the victims' relatives.
First, Margaret Gilmore reports.
</shot_text>
</shot>
<shot start="88.96" end="95.40">
<keyframe>130104/130104/130104_0005-0001.jpg</keyframe>
<shot_text>
.
.
</shot_text>
</shot>
</story>
.
.
</programme>
```

Figure 6.9: Truncated example of XML representation of ANSES data.

Figure 6.10: The front page of the ANSES interface displays the latest news.

Figure 6.11: Users can search using keywords

Figure 6.12: The search page provides a link to other news from that day. The referring story is highlighted in red.

Figure 6.13: The old ANSES interface only displayed a single key frame per story — typically an anchorperson.

With each story returned by a keyword search, a link is given to other news from the day that that story was broadcast. This links the user back to the date-based interface described above, with the referring story highlighted in red. This is shown in Figure 6.12.

Interface features common to both views

The system is designed to have a familiar look and feel to users of conventional text-based web search engines, and results are paged in divisions of 10 stories, with familiar links to navigate backwards and forwards between the pages.

An immediate visual summary of each story is provided by a series of key frames. The capture system stores one key frame for each of the shots delineated by the original shot boundary detection process, each key frame being the 15th frame of its respective shot (this usually ensures that it is clear of any gradual transition with which the shot may have started). In the interface we display up to 5 key frames: if a story contains less than five shots we display all of the key frames, otherwise we display the first, last and three roughly equally spaced key frames from in between. This multiple key frame display provides a much better visual summary of the story than was given by a previous version of the system in which only the first shot key frame was used to represent the story. The old system is shown in Figure 6.13; the single key frame was typically an anchorperson shot and thus provided little information about the story and no distinction between stories (other than the story caption, which is barely visible at the displayed resolution).

The first sentence of the story is displayed in italics below the key frames for the story, as this often in itself provides an accurate summary of the story. This is followed by the

automatically generated text summary. Alongside the text summary, the key entities are listed and colour coded according to their type — organisations, people, locations and dates, as previously described and shown in Figure 6.8. The colour scheme and fixed display location next to the summary help the user to quickly answer the critical 'who', 'when' and 'where' questions of news reports.

The user also has access to the full text of the subtitle transcript for each story — useful if they wish to fully examine a story's contents before downloading (for example if they are on a low bandwidth connection), or if they wish to find out how their text search terms were pertinent to the story that has been retrieved. The full text interface is shown in Figure 6.14, and is consistent with all other displays in the interface.

The video and associated text are stored in the database in the SMIL format [104] and the interface offers a button for each clip to be played back with its corresponding subtitles using the RealPlayer, which is available free for most platforms. The playback of a news story is shown in Figure 6.15.

User comments and advice from HCI experts have been taken into account in the construction and development of the interface but no formal user evaluation has thus far been carried out. Our intention was not to build a perfect interface, but rather to construct a system which powerfully demonstrates the value of our work in the areas of segmentation and content-based retrieval and how these can be combined in an effective video retrieval and summarisation system.

6.3 Integration with a browsing framework

In order to demonstrate the usefulness of integrating various methods of video summarisation, we have brought together the news summarisation work with some of our work on feature-centred content-based retrieval into an integrated system. The browsing interface was developed by Daniel Heesch, incorporating our earlier work on feature-based image retrieval. We make mention of it here in order to demonstrate how the feature-based work can be integrated with the ANSES system.

The front end for the system is the interface developed for the ANSES news summarisation system. The user can bootstrap a search by browsing the news for a particular day, or by entering keywords. By clicking a key frame of interest, the user opens up a new window, with the key frame at the centre of the browsing network developed by Heesch et al [34, 36], from which point they are able to find other related key frames.

An example search is shown in Figure 6.16, where an initial text-based search has been started with the keyword "Mars". In a story of interest, there is a key frame containing an image of the surface of Mars, with the NASA robotic craft "Spirit". (Incidentally, this is a powerful demonstration of the value of using multiple key frames for each story as the key frame of interest is a significant time into the story). Clicking the key frame brings up the new search window, and from this point the user can begin a graphically orientated search — potentially adding further images to the query, and browsing temporally through the

Figure 6.14: A single click from the story summary takes the user to the full story

Figure 6.15: Playback with the subtitles in the RealPlayer

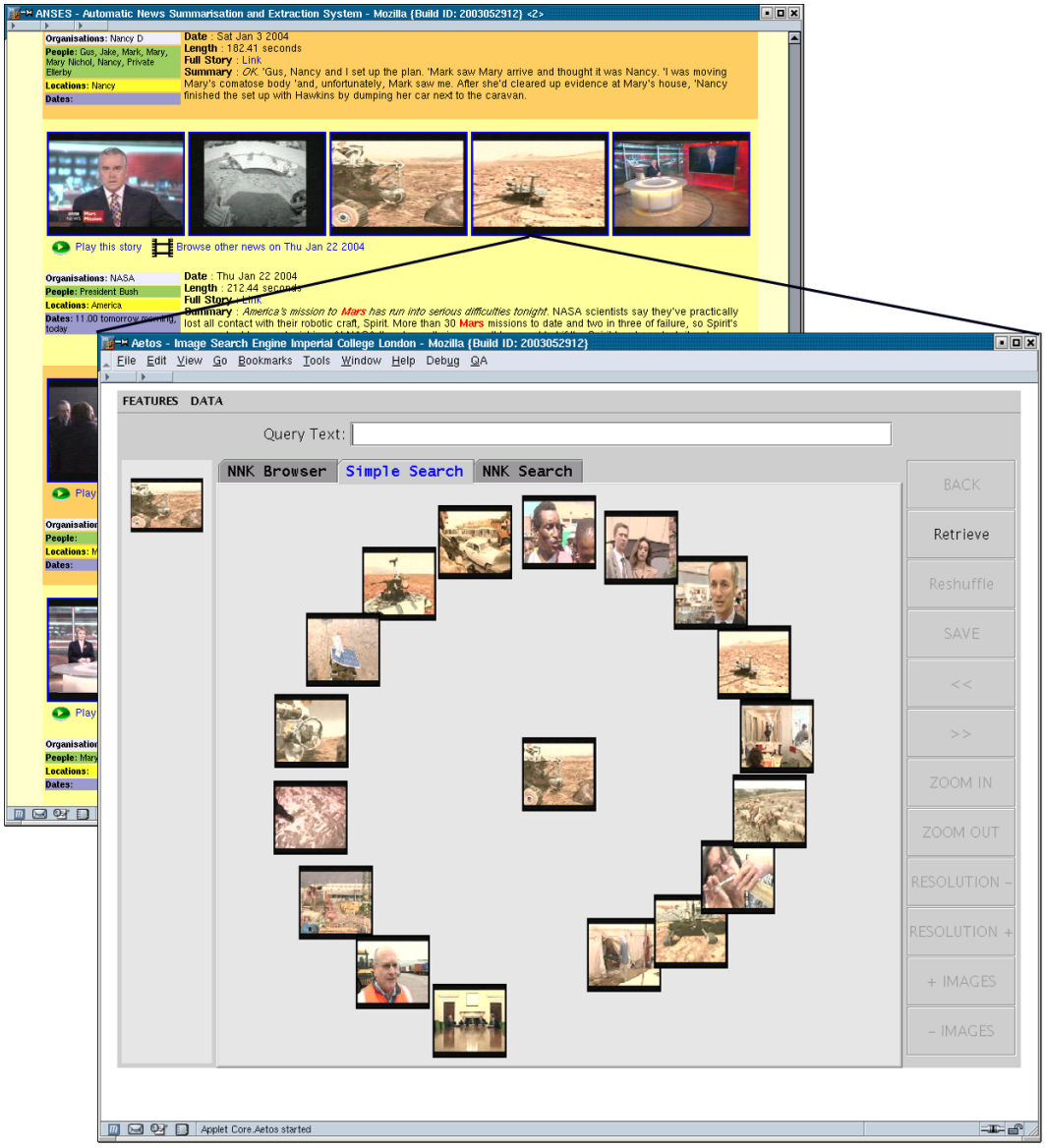

Figure 6.16: Example search using browsing interface

broadcast.

The browsing system, which was deployed to great effect in the TRECVID workshop (see Section 4.7), convincingly demonstrates the value of our work in feature-based retrieval using key frames described in Chapter 4 and shows the potential for the integration of content-based image retrieval into the ANSES system.

The growing ANSES news collection is also a useful addition to the browsing framework, since this daily growing collection serves to test the scalability of the content-based system.

Chapter 7

Conclusions

7.1 Video segmentation

Segmentation of video material into manageable pieces is important both to provide the user with a fine-grained location of material of interest, and to cope with bandwidth considerations.

Our shot boundary detection system has been shown to be highly effective, comparing extremely favourably with all other systems submitted in the latest TRECVID workshop, particularly for gradual transitions where many other systems were weak. Only two systems produced more impressive results, an advantage that was arguably outweighed by the complexity of those systems.

Our algorithm was based on video in the uncompressed domain, a constraint that was initially imposed by hardware available to us. However, as MPEG has become the standard for video compression it would make sense in the future to exploit the information intrinsically encoded in this format in order to reduce computational complexity and thus execution time.

Story boundary detection is an important process for grouping shots into more logical semantic units, particularly in the broadcast news domain where a user is likely to want to find a whole story and will be frustrated by having to view it in piecemeal fashion. The shot boundary detection system has provided a reliable basis for story boundary detection algorithms. A text-based algorithm for segment merging was implemented, as well as an anchorperson-based algorithm using the k-NN retrieval system. The text based algorithm has subjectively shown impressive results in our BBC news search system, although a formal evaluation using different data within the TRECVID framework did not fully underline this.

Our simple combination of the text-based and anchorperson-based systems did produce an improvement in precision over using either system separately, but recall was disappointing. Time limitations meant that no more sophisticated combination techniques could be employed. Better results are likely to be yielded in the future by a weighted combination of evidence, or through the use of an approach such as Bayesian networks, rather than through

the sequential approach taken thus far.

7.2 Retrieval by learning in key frames

An image collection was carefully constructed containing a wide range of images from an inexpensive and widely available database. This collection is easily reproducible for groups wishing to compare systems against those we have presented. The range of images present in the collection makes it suitable for evaluating a variety of different methods.

Having a sound test collection allowed us to perform a novel and effective comparison of three different retrieval methods (boosting, vector space model and k-nearest neighbours), using a number of different feature vectors. Despite the limitations of the Corel categories, our three retrieval methods were shown to perform well. The k-NN approach returned impressive results, even with visually disparate queries, and the boosting algorithm performed well at retrieving video key frames sharing similar visual composition. Results from this evaluation were used to great effect to inform the design of systems for the search task of the TREC 2002 video track, and for the TRECVID 2003 workshop, both of which further served to demonstrate the value of applying these methods to video retrieval.

The work at TREC and TRECVID showed remarkable results, with our system comparing extremely favourably with all other groups in the evaluation. Our approach using global features was a good generic approach which worked well across a broad range of queries, but was slightly weaker when it came to searching for specific objects or personalities. Specialist modules could be designed for these tasks in the future.

The work clearly demonstrated the potential benefits of feedback based learning techniques for classification tasks and the potential for some of the skeletal structure of a video to be actively modelled, and for familiar way points, such as anchorperson shots, to be automatically identified.

7.3 Colour classifiers

The colour classifiers have produced some promising results and the grass classifier, which was tested on vegetation at the TRECVID 2003 workshop, showed high recall performance.

There is clearly a good deal of work to be done on the classifiers, and it will be necessary to introduce discriminants such as shape, texture and contextual information in order to improve precision. Part of the problem for the algorithm as it stands is that it is dealing with isolated regions. Humans are unable to identify many regions when they are removed from their context; the next stage, therefore, is to bring together context information into a Bayesian network [65] and an implementation of Pearl's message passing algorithm [64] (which is described in detail by Yow [112]) has been partially developed.

Although this work stands alone at the moment, it was always intended that it could be integrated into a video search system such as ANSES to facilitate detailed content-based queries in which specific colours, regions or objects could be described using natural

language. This is particularly important in the absence of example images or where no transcript or other meta data exists for the video.

7.4 News summarisation and retrieval systems

The ANSES and Aetos systems impressively showcase the research described in this thesis, and bring together state of the art text summarisation techniques with our own research in boundary detection and in feature based video retrieval.

The ANSES system effectively demonstrates how shot and story boundary detection can facilitate browsing and retrieval of broadcast news, and how visual and textual information can be combined to summarise the data for easy access. A visual summary is provided through shot level key frames for each news story and this is supplemented with textual summaries derived both through extraction of key entities from the story and through a natural language summary using lexical chain analysis.

Integration of ANSES with the Aetos browser serves to demonstrate how our work on feature extraction from key frames has enabled a powerful system for content-based exploration of a visual database, and the effectiveness of the browser was proven through experiments in the TRECVID workshop. A click on a key frame in the ANSES interface takes the user into the content-based browsing interface which then enables an entirely visual search through the database.

7.5 Contributions to the literature

- Participation in TREC and TRECVID series:
  - 2001: Multi-timescale video shot change detection [70].
  - 2002: Video retrieval using global features in key frames [68].
  - 2003: Video retrieval within a browsing framework using key frames [34].
- International Conference on Image and Video Retrieval:
  - 2002: Video retrieval by feature learning in key frames [71].
  - 2003: ANSES: Summarisation of news video [72].
- Journal of Computer Vision and Image Understanding:
  - Evaluation of key frame based retrieval techniques for video [69].
- International Conference on Acoustics, Speech and Signal Processing:
  - A Comparative Study of Evidence Combination Strategies [110].
- Joint Conference on Digital Libraries:
  - Digital Library Access via Image Similarity Search [33].

7.6 Future work

It is unlikely that a great deal more performance could be squeezed from the already highly accurate shot boundary detection system, though some domain-specific tuning could be carried out in order to improve results at TRECVID. However, as we have discussed, it would be worth looking into compressed domain algorithms, even if only for the sake of efficiency, and because MPEG has become the standard for video compression and storage.

The story boundary detector has room for improvement, and the introduction of the lexical chain algorithm should make for more sophisticated analysis of text content, and there is potential for this to be combined in a more intelligent way with the output of the anchorperson detector, perhaps using a Bayesian network. It would be important in the future to look at the data and at where the current system fails, and hence what improvements could be carried out to avoid such failures. In particular it would be worth revisiting the assumption that story boundaries always correspond with shot boundaries, and at least seeing whether there might be certain exceptions to this rule which could be factored in.

The colour based classifiers need significant work to improve their precision, and would thus benefit from being combined with classifiers based on other cues such as shape and texture, as well as contextual information. Performance could also be improved through more extensive training, perhaps using the data generated in the TRECVID collaborative annotation forum. In order to facilitate detailed queries the vocabulary needs to be expanded and an intelligent method developed for the combination of evidence from the different classifiers in order to respond accurately to user queries.

ANSES is a mature and highly developed system, but would still gain from improvements in the underlying technology; users would certainly benefit from better story boundary detection, and text summarisation can be improved, though the latter is a research area all on its own. A user study of some kind may also highlight deficiencies or potential enhancements. Future developments will also include much tighter integration with the Aetos system, unifying the interfaces and maintaining the correspondence between the key frames and associated news stories. The work on global feature based retrieval using key frames needs to be supplemented with evidence from more specialist object and face detectors. Many queries require analysis of the object and camera motion content of the video and this has not yet been touched upon in our research.

7.7 Overall conclusions

Our video shot boundary detection approach has been proven to be comparable to the best systems in the field and was used to good effect as the basis for a story boundary detection system. Story boundary detection provided the logical retrieval units for content based retrieval, summarisation and browsing of broadcast news which we implemented in the ANSES system, providing web-based access to a whole year of television news.

Our evaluation of global feature based retrieval of video using key frames helped shape our design of a highly effective system which again compared extremely favourably to systems produced by large and better resourced groups from around the world. We have demonstrated the potential for this system to work within the ANSES framework, to supplement keyword-based search with search based on video content. Video key frames returned through a date-based or keyword-based search are used as query images which bootstrap the image content-based search using the browsing network. Although the global approach means that specific problems, such as face recognition, are not tackled, it has been proven to work well with a wide range of queries and could be supplemented with more specialist detectors in future.

The colour-based region classifiers that we have developed have the potential to be integrated into the retrieval system at a later stage, providing the user with the means to make detailed content-based queries using natural language and without the use of example images.

We have presented novel approaches to video shot and story boundary detection, and content-based retrieval using key frames. The techniques have been deployed in a powerful system for retrieval and summarisation of news video, bringing a solution to the video retrieval problem one step closer.

References

[1] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. In *Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA)*, Feb. 1998.

[2] A. Amir, M. Berg, S.-F. Chang, G. Iyengar, C.-Y. Lin, A. P. Natsev, C. Neti, H. Nock, M. Naphade, W. Hsu, J. R. Smith, B. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[3] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu. Content-based browsing of video sequences. In *Proceedings of ACM International Conference on Multimedia*, Oct. 1994.

[4] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain.*, 1997.

[5] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[6] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, 5(2):122–128, Apr. 1996.

[7] P. Bouthemy, M. Gelgon, and F. Ganasia. A unified approach to shot change detection and camera motion characterization. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1030–1444, Oct. 1999.

[8] M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck-Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. In *Proceedings of the Third ACM Multimedia Conference*, Apr. 1995.

[9] P. Browne, C. Czirjek, G. Gaughan, C. Gurrin, G. J. F. Jones, H. Lee, S. Marlow, K. McDonald, N. Murphy, N. E. O'Connor, N. O'Hare, A. F. Smeaton, and J. Ye. Dublin City University video track experiments for TREC 2003. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[10] L. Chaisorn, T.-S. Chua, and C.-H. Lee. The segmentation of news video into story units. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, Aug. 2002.

[11] L. Chaisorn, C. Koh, Y. Zhao, H. Xu, T.-S. Chua, and T. Qi. Two-level multi-modal framework for news story segmentation of large video corpus. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[12] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), Jan. 1995.

[13] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. In *Proceedings of the IEEE*, May 1995.

[14] M. Christel, A. Olligschlaeger, and C. Huang. Interactive maps for a digital video library. *IEEE Multimedia*, 7(1):60–67, 2000.

[15] M. G. Christel, A. G. Hauptmann, H. D. Wactlar, and T. D. Ng. Collages as dynamic summaries for news video. In *Proceedings of ACM Multimedia 2002, Juan-les-Pins, France*, Dec. 2002.

[16] M. G. Christel and A. S. Warmack. The effect of text in storyboards for video navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, UT*, May 2001.

[17] M. Cooper, J. Foote, J. Adcock, and S. Casi. Shot boundary detection via similarity analysis. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[18] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science*, 19:30–41, 1968.

[19] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia*, July 2002.

[20] A. P. de Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *Proceedings of RIAO*, Apr. 2004.

[21] D. DeMenthon, L. J. Latecki, A. Rosenfeld, and M. V. Stückelberg. Relevance ranking of video data using hidden markov model distances and polygon simplification. Technical Report LAMP-TR-067, University of Maryland, 2001.

[22] M. D. Dunlop. Time relevance and interaction modelling for information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, 1997.

[23] D. Eichmann and D.-J. Park. Experiments in boundary recognition at the University of Iowa. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[24] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[25] M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In *Proceedings of the European Conference on Computer Vision*, 1996.

[26] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkahni, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28:23–32, Sept. 1995.

[27] D. A. Forsyth and M. M. Fleck. Body plans. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[28] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[29] J. L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37(1-2):89–108, 2002.

[30] A. Hauptmann, R. V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng, N. Moraveji, N. Papernick, C. G. M. Snoek, G. Tzanetakis, J. Yang, R. Yang, and H. D. Wactlar. Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In *Proceedings of the TRECVID Workshop*, 2003.

[31] A. Hauptmann, R. Jin, N. Papernick, D. Ng, Y. Qi, R. Houghton, and S. Thornton. Video retrieval with the Informedia digital video library system. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2002.

[32] A. G. Hauptmann and M. J. Witbrock. Story segmentation and detection of commercials in broadcast news video. In *Advances in Digital Libraries, Santa Barbara, CA*, Apr. 1998.

[33] D. Heesch, M. J. Pickering, P. Howarth, A. Yavlinsky, and S. Rüger. Digital library access via image similarity search. In *Joint Conference on Digital Libraries, (Tucson, Arizona, US)*, 2004.

[34] D. Heesch, M. J. Pickering, S. Rüger, and A. Yavlinsky. Video retrieval within a browsing framework using key frames. In *Proceedings of the TRECVID Workshop*, 2003.

[35] D. Heesch and S. Rüger. Performance boosting with three mouse clicks - relevance feedback for CBIR. In *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR, Pisa, Italy)*, 2003.

[36] D. Heesch and S. Rüger. NN^k networks for content-based image retrieval. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR, Sunderland, UK)*, 2004.

[37] P. Howarth and S. M. Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of International Conference on Image and Video Retrieval (CIVR)*, July 2004.

[38] X. Huang, G. Wei, and V. A. Petrushin. Shot boundary detection and high-level features extraction for the trec video evaluation 2003. In *Proceedings of the TRECVID Workshop*, 2003.

[39] B. Huet, I. Yahiaoui, and B. Merialdo. Multi-episodes video summaries. In *International Conference on Media Futures, Florence, Italy*, May 2001.

[40] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, June 1997.

[41] International Commission on Illumination. CIE colorimetry, 1986.

[42] R. Jackson, L. MacDonald, and K. Freeman. *Computer Generated Colour.* Wiley, 1994.

[43] R. Kasturi and R. Jain. Dynamic vision. In R. Kasturi and R. Jain, editors, *Computer Vision: Principles*, pages 469–480. IEEE Computer Society Press, Washington, 1991.

[44] G. Knorr. Xawtv. http://linux.bytesex.org/xawtv/ – page checked July 2004.

[45] F. W. Lancaster and A. J. Warner. *Information Retrieval Today.* Information Resources Press, 1993.

[46] A. Large, J. Beheshti, A. Breuleux, and A. Renaud. Multimedia and comprehension: The relationship among text, animation and captions. *Journal of the American Society for Information Science*, 46(5):340–347, 1995.

[47] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):59–63, 1991.

[48] F. C. Li, A. Gupta, E. Sanocki, L.-W. He, and Y. Rui. Browsing digital video. In *Proceedings of the SIGCHI conference on Human factors in computing systems, The Hague, The Netherlands*, Apr. 2000.

[49] C.-Y. Lin, B. L. Tseng, and J. R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[50] A. Mahindroo, B. Bose, S. Chaudhury, and G. Harit. Enhanced video representation using objects. In *Indian Conference on Computer Vision, Graphics and Image Processing, Space Applications Centre (ISRO), Almedabad, India*, Dec. 2002.

[51] M. K. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Image and Vision Computing*, 17(7):513–529, 1999.

[52] B. S. Manjunath and J.-S. Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11:703–715, 2001.

[53] J. Mas and G. Fernandez. Video shot boundary detection based on colour histogram. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[54] A. Miene, T. Hermes, G. T. Ioannidis, and O. Herzog. Automatic shot boundary detection using adaptive thresholds. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[55] G. Miller. WordNet, online lexical database. http://www.cogsci.princeton.edu/~wn/ – page checked July 2004.

[56] T. J. Mills, D. Pye, N. J. Hollinghurst, and K. R. Wood. AT&TV: Broadcast television and radio retrieval. In *Proceedings of RIAO*, Apr. 2000.

[57] T. M. Mitchell. *Machine Learning.* McGraw Hill, 1997.

[58] A. Moffat. Managing Gigabytes search engine. http://www.cs.mu.oz.au/mg/ – page checked July 2004.

[59] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–43, 1991.

[60] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about Corel – evaluation in image retrieval. In *Proceedings of CIVR*, July 2002.

[61] C. W. Ng and M. R. Lyu. ADVISE: Advanced Digital Video Information Segmentation Engine. In *Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii, USA.*, May 2002.

[62] R. J. O'Callaghan and D. R. Bull. Improved illumination-invariant descriptors for robust colour object recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2002.

[63] J. Oh and K. A. Hua. An efficient technique for summarizing videos using visual contents. In *Proceedings of the IEEE International Conference on Multimedia and Expo. New York, USA*, July 2000.

[64] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1998.

[65] J. Pearl and S. Russel. Bayesian networks. In M. Arbib, editor, *Handbook of Brain Theory and Neural Networks.* MIT Press, 2001.

[66] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases, 1994.

[67] M. J. Pickering. Video archiving and retrieval. Imperial College London. http://km.doc.ic.ac.uk/video-se/, 2000.

[68] M. J. Pickering, D. Heesch, R. O'Callaghan, S. Rüger, and D. Bull. Video retrieval using global features in keyframes. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Eleventh Text REtrieval Conference (TREC-11)*, 2003.

[69] M. J. Pickering and S. Rüger. Evaluation of key frame based retrieval techniques for video. *Computer Vision and Image Understanding*, 92(2):217–235, 2003.

[70] M. J. Pickering and S. M. Rüger. Multi-timescale video shot-change detection. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2002.

[71] M. J. Pickering, S. M. Rüger, and D. Sinclair. Video retrieval by feature learning in key frames. In *Proceedings of International Conference on Image and Video Retrieval (CIVR)*, July 2002.

[72] M. J. Pickering, L. Wong, and S. M. Rüger. ANSES: Summarisation of news video. In *Proceedings of International Conference on Image and Video Retrieval (CIVR)*, July 2003.

[73] D. Pye, N. J. Hollinghurst, T. J. Mills, and K. R. Wood. Audio-visual segmentation for content-based retrieval. In *5th International Conference on Spoken Language Processing, Sydney, Australia*, Dec. 1998.

[74] G. M. Quénot, D. Moraru, and L. Besacier. CLIPS at TRECVID: Shot boundary detection and feature detection. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[75] M. Rautiainen and D. Doermann. Temporal color correlograms for video retrieval. In *Proceedings of 16th International Conference on Pattern Recognition, Quebec, Canada*, 2002.

[76] M. Rautiainen, Penttilä, P. Pietarila, K. Noponen, M. Hosio, T. Koskela, S.-M. Mäkelä, J. Peltola, J. Liu, T. Ojala, and T. Seppänen. TRECVID 2003 experiments at MediaTeam Oulu and VTT. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[77] P. Rennert. StreamSage unsupervised ASR-based topic segmentation. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[78] Y. Rubner. The earth-mover's distance as a metric for image retrieval. Technical Report STAN-CS-TN-98-86, Stanford University, 1998.

[79] G. Salton. *Automatic Text Processing.* Addison-Wesley, 1989.

[80] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith. Video OCR for digital news archives. In *Proceedings of Workshop on Content-Based Access of Image and Video Databases*, Jan. 1998.

[81] L. Schamber. Relevance and information behavior. *Annual Review of Information Science and Technology*, 29:3–48, 1994.

[82] H. Scheiderman and T. Kanade. Object detection using the statistics of parts. *Internation Journal of Computer Vision*, 2002.

[83] N. Sebe and M. Lew. Robust shape matching. In *Proceedings of CIVR*, July 2002.

[84] D. Sinclair. Voronoi seeded colour image segmentation. Technical Report 1999.3, AT&T Laboratories, Cambridge, 1999.

[85] A. Smeaton, W. Kraaij, and P. Over. TRECVID 2003 – an introduction. In *Proceedings of the TRECVID Workshop*, 2003.

[86] A. Smeaton and P. Over. The TREC-2002 video track report. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Eleventh Text REtrieval Conference (TREC-11)*, 2002.

[87] A. Smeaton, P. Over, and R. Taban. The TREC-2001 video track report. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2001.

[88] J. R. Smith and S.-F. Chang. VisualSEEk: a fully automated content-based image query system. In *ACM Multimedia*, Nov. 1996.

[89] J. R. Smith, S. Srinivasan, A. Amir, S. Basu, G. Iyengar, C.-Y. Lin, M. Naphade, D. Ponceleon, and B. Tseng. Integrating features, models, and semantics for TREC video retrieval. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2002.

[90] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[91] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Databases*, Jan. 1998.

[92] The Internet Archive. Movie Archive. http://www.archive.org/movies/ – page checked July 2004.

[93] K. Tieu and P. Viola. Boosting image retrieval. In *5th International Conference on Spoken Language Processing*, Dec. 2000.

[94] E. Toernig. AleVT teletext decoder. http://www.goron.de/~froese/ – page checked July 2004.

[95] C. P. Town and D. Sinclair. Content-based image retrieval using semantic visual categories. Technical Report 2000.14, AT&T Laboratories, Cambridge, 2000.

[96] C. P. Town and D. Sinclair. Ontological query language for content-based image retrieval. Technical Report 2001.1, AT&T Laboratories Cambridge, 2001.

[97] D. Travis. *Effective Color Display.* Academic Press, San Diego, CA, 1991.

[98] TREC-11. Video track. http://www-nlpir.nist.gov/projects/t2002v/, 2002.

[99] TRECVID. Workshop. http://www-nlpir.nist.gov/projects/t2003v/, 2003.

[100] T. Volkmer, S. M. M. Tahaghogi, J. A. Thom, and H. E. Williams. The moving query window for shot boundary detection at TREC-12. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[101] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36:697–716, 2000.

[102] E. M. Voorhees. Overview of TREC 2001. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2002.

[103] E. M. Voorhees. trec_eval evaluation report. In E. M. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2002.

[104] W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 specification. http://www.w3.org/TR/REC-smil/ – page checked July 2004.

[105] H. D. Wactlar, A. G. Hauptmann, and M. J. Witbrock. Informedia: News-on-demand experiments in speech recognition. In *Proceedings of ARPA Speech Recognition Workshop*, Feb. 1996.

[106] M. J. Witbrock and A. G. Hauptmann. Speech recognition for a digital video library. *Journal of the American Society of Information Science*, 49(7):619–632, 1998.

[107] M. Worring, G. P. Nguyen, L. Hollink, J. van Gemert, and D. C. Koelma. Interactive search using indexing, filtering, browsing, and ranking. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[108] L. Wu, Y. Guo, X. Qiu, Z. Feng, J. Rong, W. Jin, D. Zhou, R. Wang, and M. Jin. Fudan University at TRECVID 2003. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[109] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas.* Wiley, 2nd edition, 1982.

[110] A. Yavlinsky, M. J. Pickering, D. Heesch, and S. Rüger. A comparative study of evidence combination strategies. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

[111] M. Yeung, B. L. Yeo, and B. Liy. Extracting story units from long programs for video browsing and navigation. In *Proceedings of IEEE Conference on Multimedia Computing and Systems*, 1996.

[112] K. C. Yow. *Automatic human face detection and localization.* PhD thesis, University of Cambridge, Department of Engineering, 1998.

[113] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of 3rd ACM International Conference on Multimedia*, 1995.

[114] Y. Zhai, Z. Rasheed, and M. Shah. University of Central Florida at TRECVID 2003. In *Proceedings of the TRECVID Workshop*, Nov. 2003.

[115] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, 1:10–28, 1993.

[116] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4):643–658, 1997.