VIDEO RETRIEVAL USING OBJECTS AND OSTENSIVE RELEVANCE FEEDBACK

Paul Browne, B.Sc., M.Sc.

A dissertation presented in fulfilment of the
requirement for the degree of Doctor of Philosophy

Supervisor: Prof. Alan F. Smeaton

DCU

School of Computing
Dublin City University
Glasnevin
Ireland

January 2005

Declaration

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

Signed : Paul Brown
ID No. : 96164-4-68
Date : 21/01/2005

This thesis is dedicated to my parents, Jackie and Thomas Browne, thank you for your constant support.

Acknowledgements

I would like to thank my wife, Jiamin Ye Browne for all her help and support throughout my study, for her advice and willingness to review this thesis and for providing many invaluable suggestions.

I kindly acknowledge and thank my supervisor Prof. Alan Smeaton for giving me the opportunity to pursue another postgraduate degree, his guidance and suggestions during the course my work have been invaluable.

I sincerely thank my family for all their encouragement and understanding, while I spent 'a few more years' in college.

I would like to thank Tomasz Adamek for supplying me with the Simpsons object detection data and his willingness to tweak and update his code even when his own paper deadlines were looming.

I also would like to thank the following people who helped me a great deal with my user experiments:

Sorin Sav
Paul Ferguson
Bart Lehane
Jovanka Malobabić
Philip Kelly
Hyowon Lee
Neil O' Hare
Cathal Carr

Ciarán Ó Conaire
Cedric Chaise
Andy Redmond
Fabrice Camous
Mike McHugh
Georgina Gaughan
Ronan McKitterick
Sandra Rothwell

Very Special thanks to Jiamin for spending long hours helping me run the experiment, and indexing the result baseline.

Special thanks also to Cathal Gurrin and Pete Wilkins for providing me with machines to run my experiment on.

Thanks to Kieran, Jiamin and Wu-hai for their help and the afternoon coffee breaks.

Thanks to all the CDVP CA gang for all the help and fun times over the years, Kieran McDonald, Cathal Gurrin, Thomas Sodring, Hyowon Lee, Jiamin Ye, Georgina Gaughan, Neil O' Hare, Sandra Rothwell and Pete Wilkins.

Finally, I would like to thank everyone at the Centre for Digital Video Processing and the School of Computing, DCU, for providing me with a fantastic research environment.

I gratefully acknowledge the support of the Informatics Directorate of Enterprise Ireland for my studentship funding from both the IP2000 and L'OEUVRE projects.

Thanks to the generous funding of DELOS for my attendance at DELOS 2001 summer school on digital library technologies.

Abstract

The thesis discusses and evaluates a model of video information retrieval that incorporates a variation of Relevance Feedback and facilitates object-based interaction and ranking. Video and image retrieval systems suffer from poor retrieval performance compared to text-based information retrieval systems and this is mainly due to the poor discrimination power of visual features that provide the search index. Relevance Feedback is an iterative approach where the user provides the system with relevant and non-relevant judgements of the results and the system re-ranks the results based on the user judgements. Relevance feedback for video retrieval can help overcome the poor discrimination power of the features with the user essentially pointing the system in the right direction based on their judgements. The **ostensive** relevance feedback approach discussed in this work weights user judgements based on the **order** in which they are made with newer judgements weighted higher than older judgements.

The main aim of the thesis is to explore the benefit of ostensive relevance feedback for video retrieval with a secondary aim of exploring the effectiveness of object retrieval. A user experiment has been developed in which three video retrieval system variants are evaluated on a corpus of video content. The first system applies standard relevance feedback weighting while the second and third apply ostensive relevance feedback with variations in the decay weight. In order to evaluate effective object retrieval, animated video content provides the corpus content for the evaluation experiment as animated content offers the highest performance for object detection and extraction.

TABLE of CONTENTS

LIST of FIGURES

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

LIST of TABLES

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

Chapter 1 Digital Video Retrieval

1.1 Introduction

Since the growth of the Internet in the early 1990's, web search engines have proved invaluable in aiding users in their search for text documents. Without such powerful and easy-to-use search engines it is doubtful that the Internet would have become as essential in our daily lives as it has done. Today's text retrieval performance is generally very high even when the number of indexed documents is well over four and a half billion web pages and queries are only 2 or 3 words on average [Google].

Naturally there are media types other than text like cinema, photographs, paintings, statues & sculptures, radio & television broadcasts and mosaics. Visual media itself precedes text by many thousands of years, as anyone lucky enough to have visited the 34 thousand year old cave paintings in France will tell [CHAUVET]. In fact the earliest examples of basic written text are around 6 thousand years old and were found in Uruk Mesopotamia.

Despite the fact that visual media have been around for so long, current visual retrieval performance is still very poor in comparison to text. Part of the reason for this of course has to do with their creation. In order to get across ideas and concepts, written text contains highly semantic information in a pure form, with documents stored on computers being digital representations of this semantic information. The creation of documents like web pages, research papers and books requires structured text as a semantic input, as does their indexing and retrieval. Text documents stored digitally or otherwise have semantic information innately and therefore there is little effort required from computers to offer basic search and retrieval based on simple text-matching techniques.

In contrast to text documents, the creation of visual and audio-based media records a visual and audio representation of what is seen and/or heard, and the semantic meaning in a text form is not recorded. The loss of semantic meaning isn't a problem for us as our brains can reconstruct this information when the media is viewed and listened to, and our brains are doing this constantly for everything that we see and hear on a daily basis. Computers however cannot think or reason effectively and there is the problem which makes retrieval of visual-based media so hard.

It is perhaps a little ironic that while the creation of human visual and audio expressions occurred thousands of years before the first written text, it is the latter that is retrieved with such ease.

Much of our daily life is visual and audio-phonic in nature with only the smallest fraction recorded in textual format if at all. As I can attest to in the writing of this thesis, the creation of text annotation for visual and audio files is a very slow process. If one were to write (annotate) a digitised movie into textual form it would take many times the duration of the content; a conservative figure would be 20 times the duration/ real-time. It is perhaps this cost that explains why so few hours of video are manually annotated.

All things Digital

A great deal has been written about the continual growth in all things digital. Large increases in computer processing and storage coupled with people's expanding use of digital cameras and camcorders (outselling their analogue counterparts after only a few short years) it is easy to see why there is a demand for more efficient automatic retrieval of such media.

Computer hard-disk storage itself is also moving from just computer-based hardware into mass-market consumer devices like digital video recorders, games consoles, audio sound systems, Apple's iPod audio player (Figure 1-1) and digital satellite video recorders. As hard disk storage costs decrease we can expect a natural increase in the amount of digital media stored.

In the late 1980s while many people were focused on the cost of storage hardware they failed to see the need for large-scale digital video retrieval. In 1997 the average home computer shipped with a 10 gigabyte hard drive, in 2004 computers with 120 gigabyte hard drives are standard. Video compression has also improved; we can now fit 2-3 times as when we used much video into the same space as older compression technologies, but with the same quality. This means that a lot more video content can be stored on even higher capacity hard disks. At the time of writing[1] a single 300 gigabyte computer hard disk costs in the region of 235 Euro which works out at a price per gigabyte of 78.3 cent. With this level of storage capacity one could expect to store any of the following (including computer file and operating system):

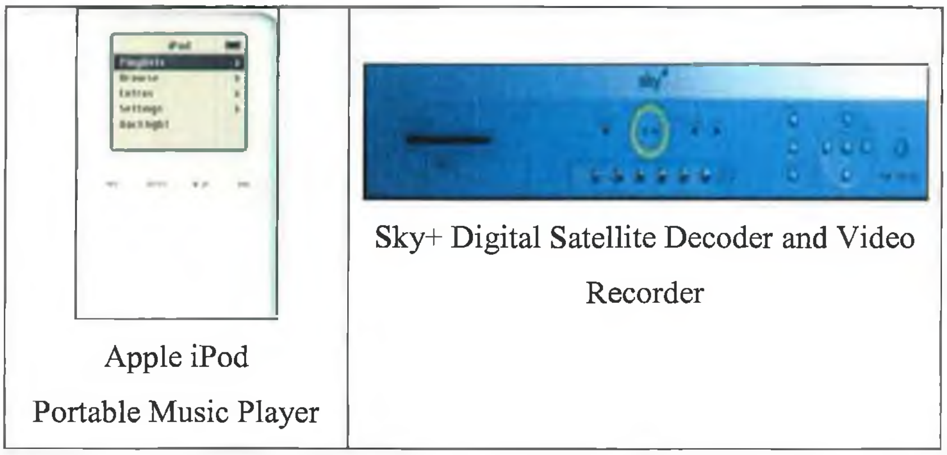
- 67,500 audio music files (mp3 full audio CD quality) while newer compression techniques from Microsoft or Apple would double that figure while still offering a similar quality level.

- 112 hours of full DVD video at near television broadcast quality.

- 575 hours of MPEG-4 video (better than VHS quality)

- 250,002 Images from a digital camera (JPEG image, 4 Megapixel camera, lossless compression)

[1] Checked on the www.komplett.ie web site, 22nd September 2004



Apple iPod
Portable Music Player

Sky+ Digital Satellite Decoder and Video Recorder

Figure 1-1: Examples of Two Consumer Products incorporating Hard Disk technology

Motion Picture Experts Group (MPEG)

The MPEG group was originally formed in the early 1990's by a consortium of companies and academic researchers. The goal of the group was to create an interoperable standard for digital video. The first video standard MPEG-1 was designed to allow digital video playback on the computers of the time.

MPEG-1 offers VHS quality video with an image resolution of 352 * 288 pixels, and the format is used for video CD. A data bitrate of 1.12 Mbps per second is typical for this format.

MPEG-2 offers TV quality video with an image resolution of 702 * 576 pixels, the format is used on DVD, digital satellite, digital terrestrial television, digital video recorders and HDTV. A data bitrate of 4.5 Mbps per second (Million bits per second) is typical for this format.

1.2 Challenges for Video Retrieval

The main goal for Video information retrieval (IR) is to achieve a similar quality of performance to text search engines on video data. While text retrieval has been around for a number decades with huge strides being made even in the last decade due to the emergence of the web, the state of video IR is still in its infancy. The following are some of the main challenges for effective video information retrieval in descending order of difficulty:

[1] Automatic object detection and extraction across domains

The search for a named person or event is an important video IR goal and one where current research offers us only a limited solution, most of the semantic information used to retrieve video is extracted from the audio information.

Objects are anything that is visible in the video, a specific person, car, trees, house, aeroplane, telephone etc. There are practically an unlimited number of real world objects that can be recorded; their detection can be automatic, manual where the user locates the object or semi-automatic where the user aids in finding an object.

Current object-based approaches tend to be domain specific and focus on retrieval of a number of pre-defined objects [Browne et al., 2003]. While the reality of this goal is ambitious and perhaps infeasible, automatic detection and search of all object types across domains remains a Holy Grail for video IR. We will discuss more about object detection in Chapter 2.2.6.

[2] A Video IR engine capable of indexing and retrieval of billions of video files

As discussed earlier, video and text are very different types of media. The size of a large novel stored digitally would still be smaller than 30 seconds of digital video even with today's very high quality compression. A video search engine would need to efficiently index and search video content that is not only larger but is orders of magnitude more complex than text. The main reason for this complexity is the number of features derivable from video which can be used for retrieval (see Chapter 2) and the level and amount of semantic information which can be extracted from the visual aspect of video. Considering the performance of text-based retrieval systems, we can expect good evaluation performance for complex retrieval tasks on large datasets [TREC 2003] [Cleverdon et al., 1966], whereas for video IR, the experimental datasets tend to be quite small [TRECVID 2003].

A few years ago storage and retrieval of a terabyte (1,000 Gigabytes) of data was considered a difficult and expensive undertaking, however over the last few years the speed of computing has increased considerably while the cost of storage as reduced (also considerably). While all this reduces the complexity of this challenge it remains a large technical challenge to facilitate large scale video search.

[3] Retrieval times within 1-2 seconds

Most people are accustomed to text retrieval response time from today's web search engines and will expect a similar speed for visual retrieval. What this means in practice is that users will expect a response time of around 1-3 seconds per query search. The longer the query response time the more chance the user will get frustrated and move on and try something else. If a user is waiting more than 10 seconds then he/she will move on to another task and abandon the search completely [Molich and Nielsen, 1990]. While video retrieval may require longer than 1-2 seconds to complete search times of several minutes are unlikely to be acceptable for most general users.

[4] Simple intuitive interface with the minimum of effort needed from the user

For many of today's technical successes, 'simplicity' has been key. This has certainly been true for web-based text search engines where the user simply needs to type in their information need in the form of search terms, and where over 75 percent of web searches are 3 words or less (see Table 1-1). Typing a number of keyword terms into a search engine to describe an information need does not prove too much of a strain for most users as the popularity of web search engines like GOOGLE shows. Table 1-1 shows the average number of search terms most people use for web-based queries, over 50 percent are 2 words or less. The data was taken from February 2003-2004 [Onestat].

Table 1-1: Web Search Term Usage

Number of Word Terms	Percentage	Cumulative Total Percent
1	19.02	19.02
2	32.58	51.6
3	25.61	77.21
4	12.83	90,04
5	5.64	95.68
6	2.32	98.0
7	0.98	98.98
8 or greater	1.02	100

A challenge for video IR would be to offer comparable levels of low user interaction requirements, however the nature of the content being so different to text means there will doubtless be more time and effort required from the user during search.

[5] Feature Extraction

The quality of video IR performance is dependent on the availability of discriminating features. A video has both visual and audio aspects and features need to be extracted from each. While we are able to extract a sizeable amount of semantic information from the audio layer we are hampered by poorly discriminating visual features. We will discuss this in more detail in Section 1.3 and Chapter 2.

In summary, the effective goal for video IR is to replicate the retrieval performance, user interaction requirement and retrieval efficiency offered by text IR.

1.3 Difficulties Improving Video Retrieval Performance

1.3.1 Video Feature Extraction Hierarchy

In this section we will discuss a hierarchy for features which can be extracted from video and what they mean for video retrieval. Features can be thought of as a core semantic representation of video content determined during indexing. A good example of a feature representation for web pages might be 'hyperlinks' which extracts the number and types of links in a document. One example of feature representation for images is the colour histogram that extracts the number of colours occurring in particular ranges.

There are three main levels of semantic feature extraction applicable to video data [Eakins, 1999].

Low Level Features

These are features that contain a low level of semantic information but generally operate across different domains. Used individually they often offer limited performance benefit for retrieval but this is not always the case and can depend on the particular query topic/ information need as we will see in Section 1.5.4. Colour and edge-based histograms, video motion and colour averages are all examples of low-level features.

Medium Level Features

These features contain a limited amount of semantic information. They are usually concept-based (car, plane or building detection for example) and operation can be limited to specific genre domains like television news or sports depending on the concept. Mid-level features are an attempt to bridge the semantic gap between low and high level features.

High Level Features

These features contain a large amount of semantic information. They operate across domains and offer very good retrieval performance. An example of a high level feature for video is ASR, where human spoken audio in a video sequence is converted to text.

Feature Hierarchy Example

For this example to illustrate a hierarchy for features, we will discuss three possible human face detectors showing what level they each belong to and why.

A Face Detection feature that identifies if an image contains face(s) or not is an example of a low-level feature. The reason for this is that the level of semantic information is small, and a search using this feature alone would be too vague and would probably return too many results.

A Face Detection feature that detects if an image contains person X's face would be an example of a Medium level feature. This contains more semantic information than above but is limited as it can only find specific faces, i.e. person X and if you wish to retrieve a different face then it will not be useful.

Suppose that a face detection feature existed that detects if an image contains any given and named human face then that would be an example of a high level feature. In this case we are not limited to vague results as in the first example nor are we limited in the face that we can search for.

1.3.2 Indexing features

To index a text document using a basic approach one simply removes commonly occurring words known as stopwords, examples include (like a, are, my, and, the, will, I, you) etc., and document indexing is based on the terms that remain. Search and retrieval can simply return documents that contain matches to words that the user has inputted. The level of processing required for this basic example is quite small, and while more advanced retrieval techniques require more processing, still the performance quality can be quite high [TREC 2003].

As we discussed earlier visual content offers a much greater challenge for extracting meaningful semantic information than text documents, the main reason for this being that text is generally created and indexed from explicit semantic information whereas the semantic information in video content is implicit and knowledge/ content-based. With video, the semantic information is implicit for the visuals and occasionally explicit in the spoken text. Indexing video even at the most basic level requires a great deal more processing time and effort in order to extract the most limited implicit semantic information (i.e. levels of colours in the images). Search and retrieval on this visual information will still be very poor in comparison to a simple text indexing approach. Video indexing approaches tend to use a number of features extracted from the audio and visual layers, mostly high-level and low-level features, and no single feature alone is sufficient for high retrieval performance.

1.3.3 Human Ambiguity in Video Annotation & Search

While the performance of automatic visual indexing technology remains poor there will continue to be a need for humans in the indexing/ annotation process. Using people to index visual content does offer the prospect of very high performance but the downside to this is the higher cost involved both in time and money.

Another downside is that people themselves rarely agree on anything and this can lead to content being indexed differently depending on who is doing the indexing, when it was done and where it was done. Rigid methodology and training as well as a closed indexing vocabulary can reduce this problem somewhat but human nature being what it is 'some error is inevitable' and impossible to remove completely.

Human perceptions are different, how we view the world around us depends on a large number of factors like where we were born, family, friends, religious customs etc. All these factors will cause people's reactions to the same stimuli to vary.

Another contributing factor towards ambiguity in human video annotation is the video content itself, which can be ambiguous in nature, and belong to more than one distinct category. This makes indexing the content more difficult and prone to error. Annotation cannot cover all types of media equally as each has its own particular attributes that need to be considered, nor can it be expected to fully annotate any media.

Table 1-2: Possible Image Annotation

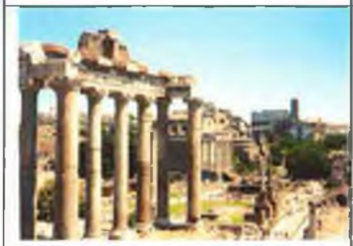
Image Example	Possible Annotation
	Ancient Rome, Senate Building Foreground, Roman Coliseum, Roman Baths, Blue Sky Background. Unidentified People Background, Trees Recorded: 14:00 hours, 12th April 2004

Table 1-2 illustrates a possible image annotation and took 2 minutes to complete. When we consider that people with digital cameras are taking 100's of images while on holiday the difficulty of the annotation task becomes clear.

The annotation itself seems quite complete, however there are a number of ancient buildings visible but only three actually named. In addition, an image search for a cloudless sky would fail to find this image as the nature of the blue sky is not stated. Finally the annotation does not give an indication of the quality of the image.

1.4 Real User Testing and Evaluation

An important requirement for effective video IR is proper evaluation which is needed in the creation, modification and evaluation of retrieval algorithms, system interaction, quality of ranked results, weighting/ judgement of content and the modification of search options.

There are two main forms of evaluation: automatic and interactive. In an automatic evaluation process, the user issues a query on a once-off basis and the system returns ranked items relevant to the query. In an interactive evaluation process, a user is allowed to interactively modify their queries and browsing based on the relevant results from the previous searches.

The interactive evaluation of digital video systems is difficult since the relevance judgements of users cannot simply be automated during evaluation. If that were the case it would make testing a lot easier. In this thesis we will focus on interactive evaluation in which real users are required to complete the study.

The first stage in user evaluation is that specific content is selected for retrieval development and evaluation (documentary or television news for example) and a select number of query topics (search tasks) are also decided on. These are both very important and need to be decided with great care, the content needs to be diverse enough and in sufficient quantities, while the query topics need to be constructed carefully and in sufficient numbers to fully test performance. Query topics also need to be decided in such a way as to reduce performance bias from specific types.

The second stage tests ***N*** users on the speed and accuracy in which they identify valid results with selected retrieval system variations. As the number of users tested increases so too does the statistical reliability of the results, but unfortunately the experimental time and cost required also tends to increase and there is a limit to how much time and effort we can expect from a user in performing interactive searches. An increase in the number of different system variations that need to be tested results in an even larger increase in the number of users that will be required, so evaluations tend to be on very specific aspects of the retrieval methods. The following are the main limitations for user testing that needs to be balanced:

- Number of users available for the experiment: it can be difficult to get a sufficient number of test users;
- Amount of experimental system variations that can be evaluated: less users available mean that less variations can be tested;
- Time/scope for the experiment: more query topics mean that longer experiments are needed.

All of these need to be balanced against the overall cost of running an experimental evaluation.

1.5 Video Terminology

In this section we will describe the physical structure of digital video and discuss some of the more common terms used in the field of digital video retrieval. These will be important later in the thesis.

1.5.1 What is Digital Video

As we can see from Figure 1-2 video content is constructed from a number of continuous images usually with an associated and synchronised audio track. Images are captured/ recorded at a specific rate in order to give the illusion of motion; a minimum of 20 frames per second is needed in order to give the illusion of smooth motion and for anything less than that, the movement appears unnatural.

Audio is also captured at a specific rate with a higher rate of capture giving better quality. Capturing 3,300 audio samples per second is approximate to telephone quality audio while 44,100 samples per second is the level of capture needed to achieve CD quality audio.

When all this information is compressed and stored in a digital format of 1s and 0s then it is known as digital video. This is different from VHS video in which the video content is stored on magnetic tape in analogue format (i.e. overlapping sine waves).

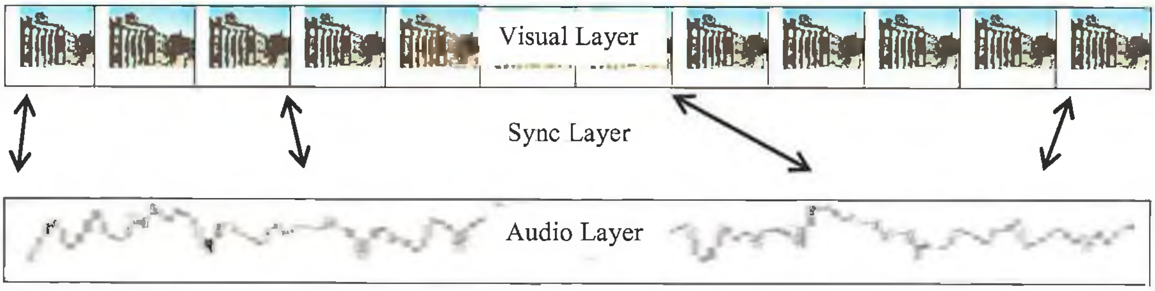


Figure 1-2: Physical Digital Video Layers

Over the last few years the importance of storing a semantic layer of description of the content of digital video has been realised. The semantic layer contains metadata information about the content of the video or audio files, for example a semantic layer in the ID tag from mp3 audio-based content (known as ID version 3 or ID3 for short). Here metadata information like artist, song title and genre information is added to the audio file and with compatible players this information can be displayed during playback.

MPEG-7 [MPEG-7] is an extendable and interoperable semantic layer created for multimedia content description and is becoming increasingly popular for video retrieval [Ye and Smeaton, 2003]. In Chapter 2 we will discuss video feature extractors that can use this layer to store their information in a structured way. Currently the MPEG-7 layer is stored separately to the content but in the future might be combined with the visual and audio data.

1.5.2 Selected Video Terminology

Frame: This is the most basic unit in video and describes a single recorded image. A number of these frames are shown per second (fps) to give the illusion of motion, popular fps values are 25 and 29.97.

Shot: A Shot can be described as continuously recorded images from a single camera source in time. As an example, imagine we were watching a news programme with the anchorperson describing the current story then the camera showed a different person in the studio talking or indeed the same anchorperson from a different angle, that would be a change in shot.

Shot Cut
When a shot changes there is a shot cut, and this can be ‘hard’ (occurring between adjacent frames) or gradual (like a fade or dissolve) occurring over a number of frames. Shot changes can be detected automatically very accurately, usually to within an accuracy 85%-95%.

Scene
A scene is a group of semantically related shots. Naturally scene detection is a far more difficult problem than shot cut detection and is not a solved problem as the shots in a scene might be visually quite dissimilar making their detection a challenge. Depending on the domain of video content some scene detection is possible (i.e. TV news programmes), as we will see in Section 1.6.3.

Keyframe
This is a single image used to visually represent a shot. Finding an appropriate keyframe to represent a shot is a difficult task as we are talking about something that is subjective. Best empirical practice seems to indicate that the middle frame of a shot be used as the representative keyframe thereby avoiding neighbouring shot transition frames. This works fine in most cases but long shots may need more than one keyframe in order to be represented visually.

Fade-out
This is a type of shot transition where n frames from the end of a shot gradually fade to black or in rare cases to white. The full black or white frames become the next shot

Fade-in
Following on from the fade-out this shot transition takes a black frame and fades it into the nth frame of the next shot. Fade-in and Fade-outs and be detected automatically quite accurately.

Dissolve This is a more complicated shot transition, where n overlapping frames from two neighbouring shots are gradually dissolved into each other by some film editing ‘process’. There are many different types of dissolve, an example of a simple one takes blocks (i.e. evenly divided regions in an image frame) from the next shot and gradually adds more and more of these to the end of the earlier shot.

Video Layers Video is actually made up of a number of layers, currently three. The first is the *Visual layer* which takes up most of the space, the second is the *Audio layer* and the final layer is the *System layer* (used to synch the audio and video). A Semantic Information layer is sometimes available (like MPEG-7 for example) currently stored separately from the video file but this will change in the future.

1.6 Selected Visual Retrieval Systems

In this section we will look at three different visual retrieval libraries and describe their features, interaction, goals and history. Both feature various forms of video retrieval with different search features and interaction.

1.6.1 The Físchlár Digital Video Library

In 1998, the Centre for Digital Video Processing (CDVP) in Dublin City University first started work on developing the Físchlár digital video library in order to showcase and evaluate research from within the research group. The first system developed by the group was Físchlár TV which was a collaborative online digital video recorder with shot level browsing and playback [O'Connor et al, 2000]. Users could schedule television programmes to be recorded and, when available, use various browsers to search through and playback that content. The system specialised in browsing of video content and there were a number of specialised browser variations that were created and evaluated [Lee et al., 2000].

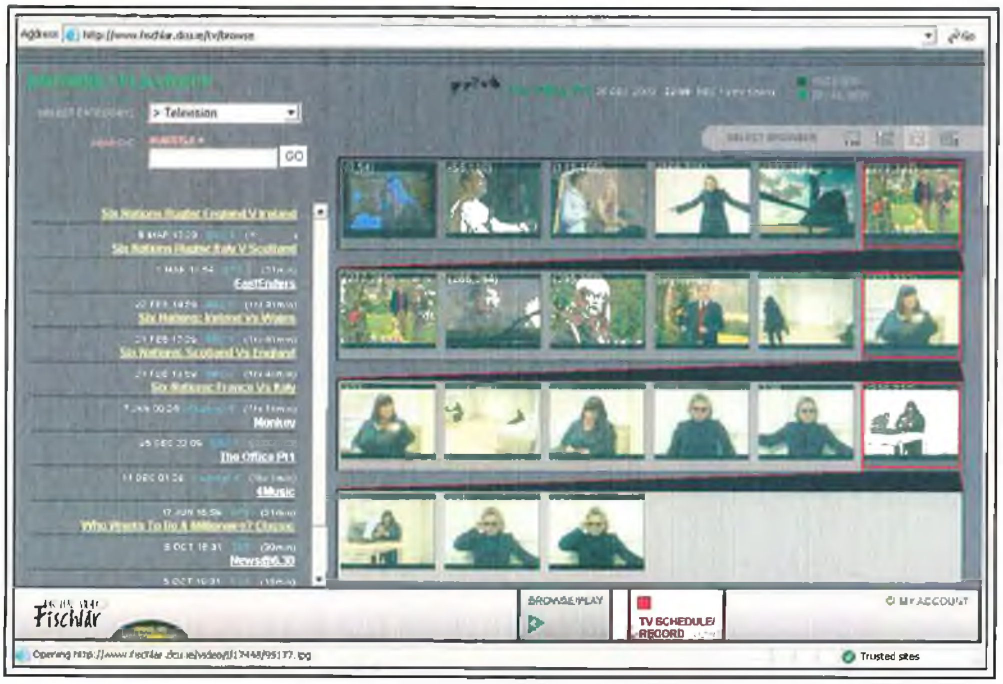
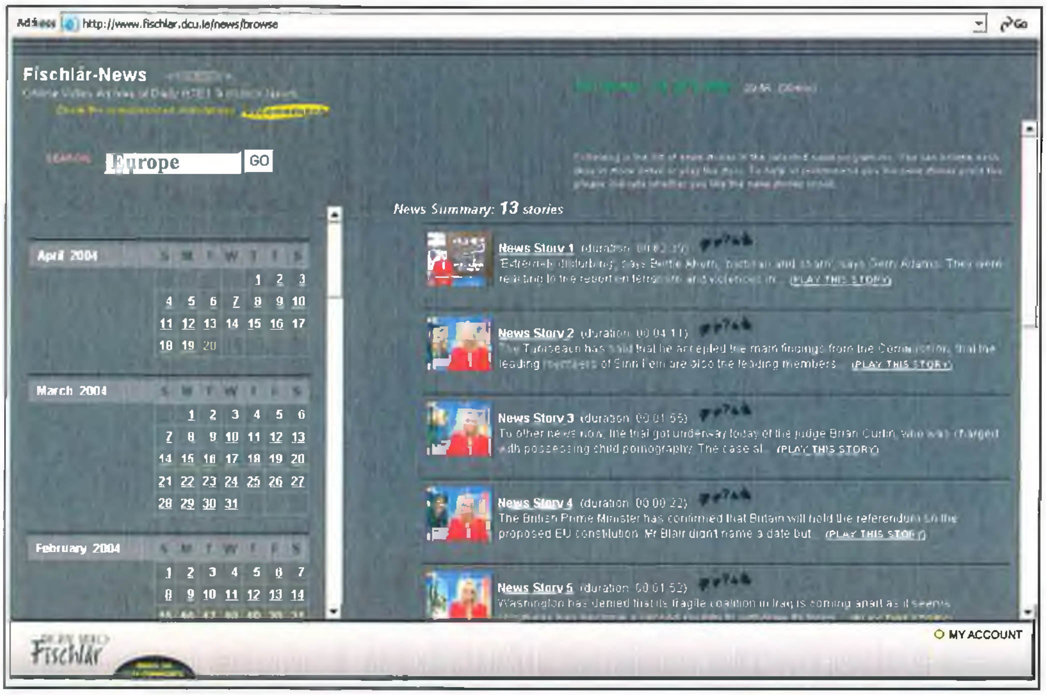

Address http://www.fischlar.dcu.ie/tv/browse
Go
> Television
GO
Six Nations Rugby: England V Ireland
Six Nations Rugby: Italy V Scotland
EastEnders
Six Nations: Ireland Vs Wales
Six Nations: Scotland Vs England
Six Nations: France Vs Italy
Monkey
The Office Pt1
4Music
Who Wants To Be A Millionaire? Classic
News@6.30
Físchlár
BROWSE/PLAY
TV SCHEDULE/ RECORD
MY ACCOUNT
Opening http://www.fischlar.dcu.ie/video/l17448/95177.jpg
Trusted sites

Figure 1-3: Screenshot of Físchlår TV

The Físchlår TV system is broken down into two main sections namely Browse/Play and TV Record. The screenshot shown in Figure 1-3 shows the Browse/Play section. The scrollbar on the left shows a list of the TV programmes recorded and available for browse and playback. Clicking on any will give the user an overview of the programme while clicking on detail view (top right) gives the user the option to browse the programme. A number of content browsers are available and the one shown above is known as the hierarchical browser.

Another system developed within the CDVP was the Físchlår News library, designed to showcase the group's domain-specific research on television news content. An early version of the system recorded and made available each nightly news programme with content stretching back over many months. Users could search the closed captions extracted from the news in order to locate content of interest. As the system evolved, fully automatic news-story detection was incorporated. Users could then browse news programmes at the story/scene level [O Hare et al, 2004].



http://www.fischlar.dcu.ie/news/browse

Fischlár-News

Europe GO

April 2004	S	M	T	W	T	F	S
					1	2	3
	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18	19	20				

March 2004	S	M	T	W	T	F	S
		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31			

February 2004	S	M	T	W	T	F	S
	1	2	3	4	5	6	7
	8	9	10	11	12	13	14

News Summary: ***13*** *stories*

News Story 1 (duration [illegible]) "Extremely disturbing", says Bertie Ahern, "bizarre and sham", says Gerry Adams. They were reacting to the report on terrorism and violence in ... (PLAY THIS STORY)

News Story 2 (duration 00:04:11) The Taoiseach has said that he accepted the main findings from the Commission, that the leading members of Sinn Fein are also the leading members ... (PLAY THIS STORY)

News Story 3 (duration 00:01:55) To other news now, the trial got underway today of the judge Brian Curtin, who was charged with possessing child pornography. The case al... (PLAY THIS STORY)

News Story 4 (duration 00:00:22) The British Prime Minister has confirmed that Britain will hold the referendum on the proposed EU constitution. Mr Blair didn't name a date but ... (PLAY THIS STORY)

News Story 5 (duration 00:01:52) Washington has denied that its fragile coalition in Iraq is coming apart as it seems

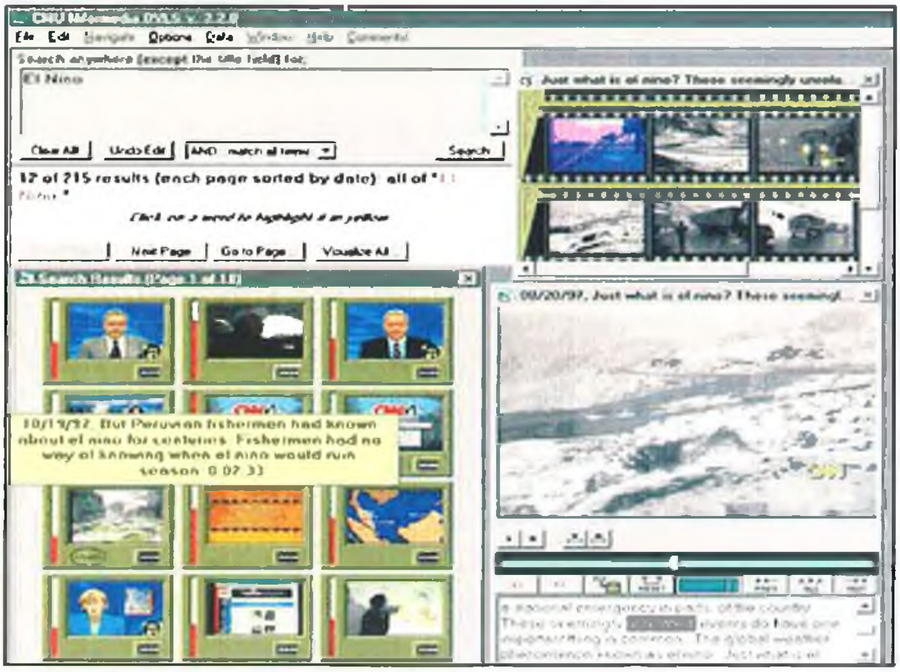
Figure 1-4: Screenshot of Físchlár News

We can see from the screenshot in Figure 1-4 that the available news programmes organised by calendar are visible on the left with the stories shown on the right. The user can start playback of the story or view the story's associated shots and text. Users can search the closed-caption text by adding terms to the search box shown on the top left of the screen.

A final variation of the Físchlár system is the Físchlár Nursing library that was created to aid the university's expanding nursing school to deliver and tailor video content for students in an interactive and visual manner. It differs not only in the video content available, which is nursing specific, but also in the approach used to index the content. Lecturers essentially created the index for the video content they wished and it was incorporated into the library. Nursing students could view the content whenever they have time available [Gurrin et al., 2004]. Nursing material is a natural choice for a digital video library due to the visual nature of the educational material.

1.6.2 The Informedia Digital Library

One of the early examples of an online digital video library was the Informedia project at Carnegie Mellon University (CMU). The first Informedia digital video library was started in 1994 and was designed to integrate speech, image and language to facilitate the construction and search of a digital video library. It ran from 1994 to 1999 with the video content mainly from governmental education material and television news (see Figure 1-5). Using the CMU Sphinx speech recognition system they automatically convert the audio to text using automatic speech recognition (ASR) and create unique methods of browsing and enhanced playback.


File Edit Navigate Options Data Window Help Comments
Search anywhere (except the title field) for:
Clear All
Undo Edit
AND match all terms
Search
Next Page
Go to Page
Visualize All

El Nino

12 of 215 results (each page sorted by date) all of "El Nino"

Click on a word to highlight it in yellow

Search Results (Page 1 of 18)

Just what is el nino? These seemingly unrela...

09/20/97, Just what is el nino? These seemingl...

10/19/97. But Peruvian fishermen had known about el nino for centuries. Fishermen had no way of knowing when el nino would ruin season 0:07:33

a national emergency in parts of the country. These seemingly [illegible] events do have one important thing in common. The global weather phenomenon known as el nino. Just what is el

Figure 1-5: Screen Grab of the Informedia interface[1]

Informedia 2 is the title for the current project and focuses on retrieval and summarisation of television news-based content. Their archive of news material stretches back to middle of the 1990's and provides a large corpus for evaluation. In the second version of the project they incorporate temporal and geographical location search from the ASR and users can navigate video content via a map of the world and a timeline bar to retrieve content of interest. The group have also

[1] Taken from the Informedia Home Page at: http://www.informedia.cs.cmu.edu/

developed a very sophisticated face-matching system and this is incorporated into the indexing and retrieval process [Wactlar, 2001] [Wactlar, 2000].

1.6.3 The CueVideo System

The CueVideo system was started as a project from within IBM's Almaden Research centre in 1997, and concluded in 2001. It was developed to address the need for an automated method of video indexing and provide improved methods for browse and search. The researchers recognised that the index and retrieval were key areas for video retrieval.

CueVideo incorporated and enhanced IBM's Via Voice speech recognition technology for automatic audio to text extraction (ASR) and developed a number of novel browse and playback tools for video retrieval. Their system was developed in two main modules, the first was the indexing and server utility that took content in a variety of formats, created an index and made this available over the Internet. The second module was a browse and search application that facilitated the users search and retrieval over the available indexes [Amir et al, 2001] [Niblack et al., 2000].

1.6.4 Selected CDVP Media Projects

The following are a number of media projects from the *Centre for Digital Video Processing* [CDVP].

One project *MediaAssit* is developing tools for digital image organisation, with the growth of digital cameras (outselling their analogue counterparts) the need for improved photo organisation is clear [CDVP].

A second project *L'OEUVRE* is developing techniques for automatic indexing, browsing and linking of digital video information. Object detection is one of the main features under development in this project [CDVP].

The third project *Adaptive Information Cluster* is developing software that can filter and personalise large amounts of digital information. A large amount of information from the Internet to Mobiles can be personalised for people's specific interests [CDVP].

The fourth project is *Físchlár on a PDA* looks at the browsing of digital video on small portable devices like the iPAQ [CDVP].

1.7 TREC Evaluation

In earlier sections we talked about the importance of proper information retrieval evaluation and in this section we will discuss the main forum that facilitates this: **TREC** or the **T**ext **R**etri**E**val **C**onference.

1.7.1 A brief history of TREC

In 1992 just as the Internet was starting to take off a US government organisation NIST (National Institute for Standards in Technology) in the United States recognised the need for a common forum for text retrieval evaluation and formed TREC. It started as part of an evaluation for DARPA's TIPSTER [Harman, 1992 B] program with 25 groups participating in the first year. The first year of TREC featured two tasks: ad hoc retrieval and routing on 2 GB of text with 50 query topics. At that stage it was a big challenge for many groups to store the 2 GB dataset.

As the years went web retrieval, information filtering and natural language processing tasks (tracks) were added. Audio-based tracks were introduced in 1995 while the first video track started in 2001. The datasets and topics for these tracks have been continually increasing in size and complexity. In terms of complexity the 1992 TREC evaluation dataset was 2 Gigabytes in size, by 2004 this had grown to over 1,000 Gigabytes for the Terabyte track.

1.7.2 TREC Video Evaluation 2001

The first year of TRECVID (the name given to the TREC video track) in 2001 had three main tasks, automatic shot boundary detection (SBD), an interactive task where results were from real user searches on predefined topics and the final task was automatic (system) based results for predefined topics. 10 groups from around the world participated in 2001. Each group sent NIST their topic run results for each of given topics, which were then pooled, with overlapping results validated manually by human assessors.

The corpus in 2001 consisted of 11 hours of MPEG-1 video (VHS Quality) from the NIST Digital Video Collection Vol-1 (Selected NIST projects from the 1980s and early 1990's) and selected videos from the open-video collection [OPEN VIDEO] (American Government documentaries from the late 1980's and 1990's). The evaluation subset of the video corpus was 7.2 hours in duration.

Overall there were 45 topics of which a subset was used by each of the three main tasks. An ASR was donated by one of the TREC groups. Table 1-3 shows the topics for the interactive task:

Table 1-3: TREC 2001 User-based Search Task Query Topics

1	Statue of Liberty showing spikes	12	Images of Lou Gossett, Jr
2	The planet Jupiter	13	All other pictures of R. Lynn Bonderant
3	Astronaut driving lunar rover over lunar surface	14	Scene from Star-Wars with R2D2 and 3CPO
4	Corn on the cob	15	Biplane flying over a field
5	Deer with its antlers	16	Sailing boat on a beach
6	Airliner landing	17	Hot air balloon in the sky
7	John Deere tractor	18	Governmental buildings looking like Capitol
8	Lunar rover from Apollo missions	19	Water Skier behind a speed boat
9	Pictures of Ron Vaughn, President of Vaughncraft	20	Chopper landing
10	Pictures of Ronald Reagan speaking	21	Additional shots of white fort
11	Pictures of Harry Hertz	22	Ronald Reagan reading speech about Space

During experimental evaluation from within the DCU group [Browne et al., 2001] many users found difficulty finding valid answers to topics 7, 11 and 12. The main reason for this is that the textual ASR which was the primary means for user search, failed to mention the query topic making search a hit and miss affair. We mention this here to motivate the need for the occasional use of actual objects which appear in the video, as part of retrieval and browsing and we shall come back to this point later in the thesis.

1.7.3 TRECVID Evaluation 2002

The second year of TRECVID featured a larger video corpus and additional video segmentation features for evaluation. The video corpus consisted of 72 hours of American 'educational' material from the 1940's to the 1960's; 42 hours were used for evaluation with the remaining 30 hours used for training and development.

A new video feature detection task was introduced in TRECVID in 2002 with evaluation of 11 different features (Table 1-4). The ASR feature was extracted from the audio, converted into text and aligned at the shot level. The word error rate for the ASR was approximately 65% [TREC IBM 2002] and was supplied by LIMSI [Gauvan et al., 2002]. The ten remaining features were all mid-level concept-based features and were supplied with confidence values ranging from 0-100. Note that the *text onscreen* feature did not supply actual text it simply identified if text was displayed on screen.

Table 1-4: TREC 2003 Features

Num	Feature	Num	Feature
1	Outdoor	7	Text Overlay
2	Indoor	8	Speech
3	Face	9	Instrumental sound
4	People	10	Monologue
5	Cityscape		
6	Landscape	*	ASR Text

Any participating TRECVID group which did not take part in the features detection task could avail of donated features supplied by a number of the TRECVID groups for their interactive task. Figure 1-6 shows a screenshot from the interactive TRECVID 2002 system developed by the DCU group [Browne et al., 2002].

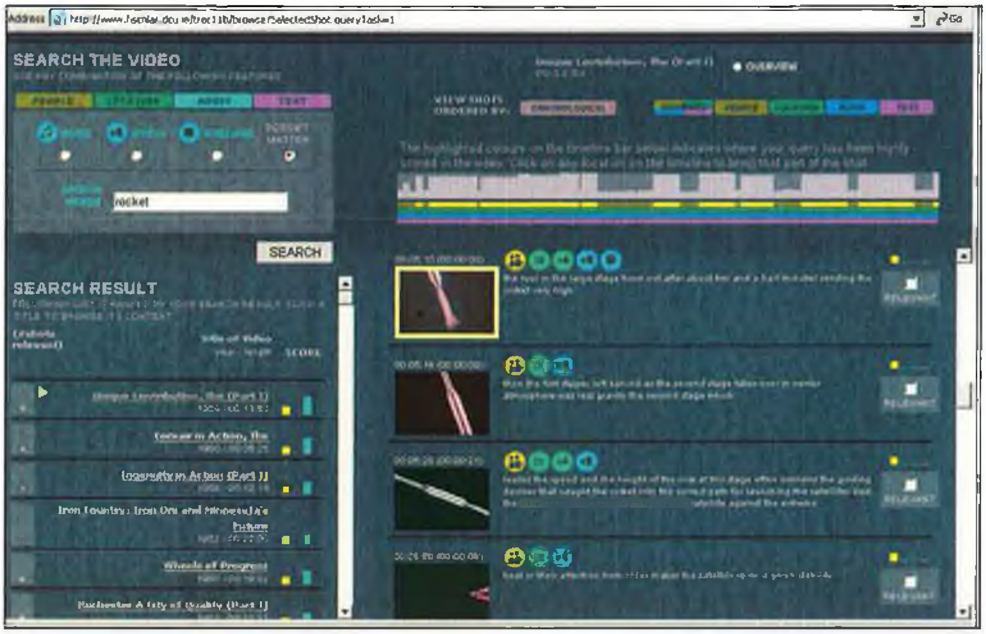


Figure 1-6: Screenshot of DCU's TREC 2002 System

The main finding from groups participating in TRECVID 2002 was that the additional features (except naturally the ASR text) failed to improve overall retrieval results. A number of reasons given for this included the age and type of the video content and the features' poor discriminating ability. Figure 1-7 shows the poor accuracy of the features (average correct 33%), which shows the difficulty of the task. The only feature that offered real performance benefit was the ASR text, and in general, the remaining TRECVID 2002 concept-based features offered limited benefit. The video collection and features were changed again in 2003 in an attempt to allow a fairer attempt at evaluating the usefulness of features in video retrieval.

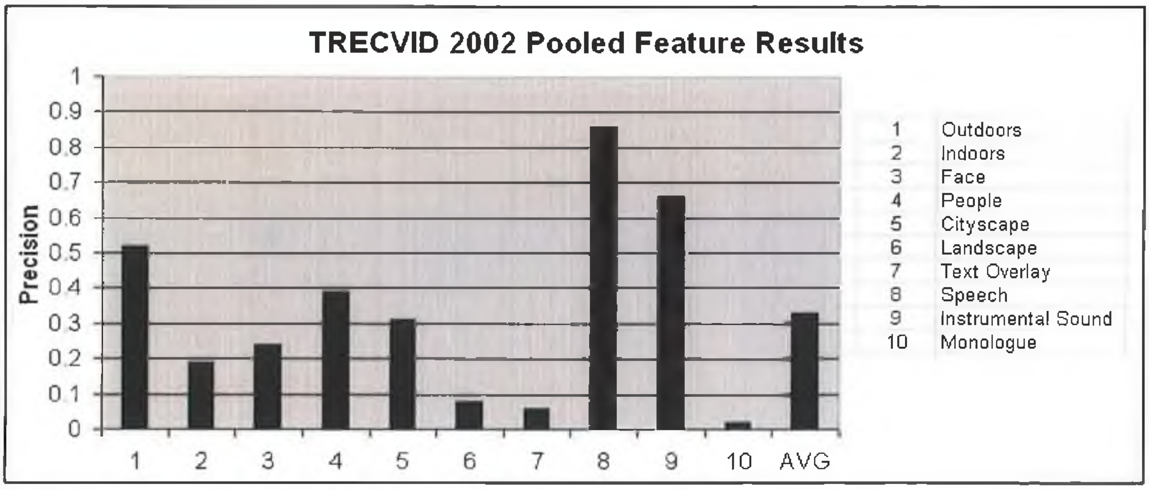


Figure 1-7: TRECVID 2002 Pooled Feature Results

1.7.4 TREC Video Evaluation 2003

The big change in TRECVID 2003 from previous years was the domain of the video used for evaluation, where 120 hours of television news content from CNN and ABC from 1998 and 13 hours of C-SPAN video was used for the development and evaluation corpus [Browne et al, 2003]. One of the reasons that television news content was chosen is its highly structured nature making story/scene detection feasible.

Apart from shot boundary detection, the 2003 TRECVID featured a news story (scene) detection task and introduced a number of new segmentation features (see Table 1-5). There are a number of domain-specific features like *news subject monologue* and *weather news* included in 2003 whereas TREC 2002's features were not domain-specific.

Table 1-5: TREC 2003 Features

Num	Feature	Num	Feature
1	Outdoors	10	Aircraft
2	News Subject Face	11	News Subject Monologue
3	People	12	Non-studio Setting
4	Building	13	Sporting Event
5	Road	14	Weather News
6	Vegetation	15	Zoom in
7	Animal	16	Physical violence
8	Female Speech	17	Person X
9	Car / Truck/ Bus		

The retrieval results showed that these additional features did boost the overall system performance for a few of the participating TRECVID groups [TRECVID 2003] despite the fact that average feature performance was similar to the previous year (Figure 1-8 & Figure 1-7). The domain-specific nature of some the features and quality of the evaluation content might be factors in the performance benefit obtained by a number of the groups but no detailed analysis of this has been reported at the time of writing. None of the features from the previous year were specific to the domain whereas in 2003 there was 5 namely *"sporting event"*, *"weather news"*, *"non-studio setting"*, *"news subject monologue"* and *"news subject face"*.

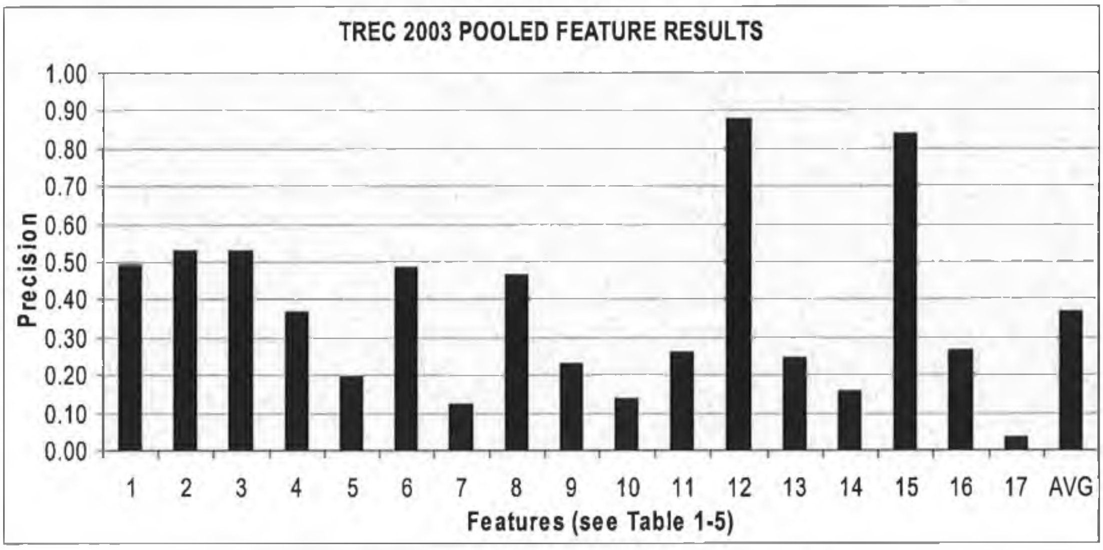


Figure 1-8: TREC 2003 Pooled Feature Results[1]

Table 1-6: TRECVID 2003 Query Topics

1	Aerial views *	13	Basketball Matches
2	Baseball Matches	14	Yasser Arafat
3	Aircraft Taking off	15	Helicopters in the Air
4	Tomb of the Unknown Soldier	16	Missile Lunch
5	Mercedes Benz logo	17	Tanks
6	Person diving	18	Train coming towards camera
7	Fire	19	Snowy Mountain top
8	Osma Bin Ladin	20	Traffic
9	Egyptian Sphinx	21	People on the Street
10	Congressman Mark Souder *	22	Actor Morgan Freeman *
11	Cup of coffee *	23	Video of Cats
12	The Pope	24	The White House
* Indicates no text found in ASR			

[1] Results Taken from TRECVID An Introduction [TRECVID 2003]

Another factor in the way that the use of features helped improve retrieval performance (especially low level features) was the nature of particular topics and/or their associated examples. Figure 1-9 illustrates two interesting topic examples that offer improved TREC 2003 retrieval performance when using features [Browne et al, 2003]. The first topic example "*Basketball Matches*" illustrates TREC topic examples that are taken from a similar video source to the content we expect to retrieve from. The second example "*Aircraft Taking off*" is a good topic for low-level features like the colour histogram as the examples contain a large amount of a dominant colour, in the example's cases it is the background blue sky which takes up a significant portion of the image.

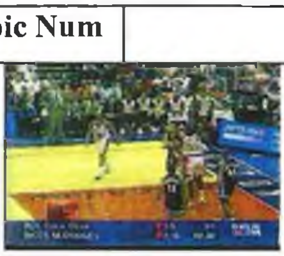
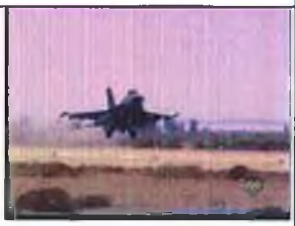
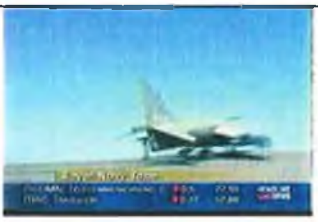
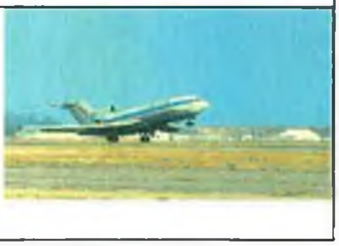
Topic Num	Topic Example Images		
2			
3			

Figure 1-9: Friendly Image Topic examples from TREC 2003

Some TRECVID query topics and their examples did pose a difficult challenge for retrieval. The following are the main reasons why some topics prove more challenging for retrieval than others:

1: The ASR audio did not contain the topic keywords (see Table 1-6). Naturally without matching keywords, text-based search offers limited performance. In the case of the query "*Congressman Mark Souder*" one solution that was tried was text-based searching on Congressman and Senator to narrow down the search and then browse through the results.

2: The TREC features were not useful for the particular query. If there was person detection feature for "*Congressman Mark Souder*" then the topic search would have been far easier, unfortunately none of the remaining features were suitable for that specific topic.

3: The Query topic examples are visually dissimilar from the search content. Retrieval systems that use low-level features and incorporate the topic examples into search would not have much success. The main reason is that the topic examples and valid results from the search content are too different in terms of colour and shape to be ranked highly and retrieved (see Figure 1-10).

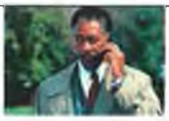
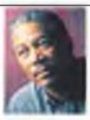
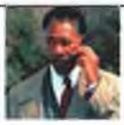
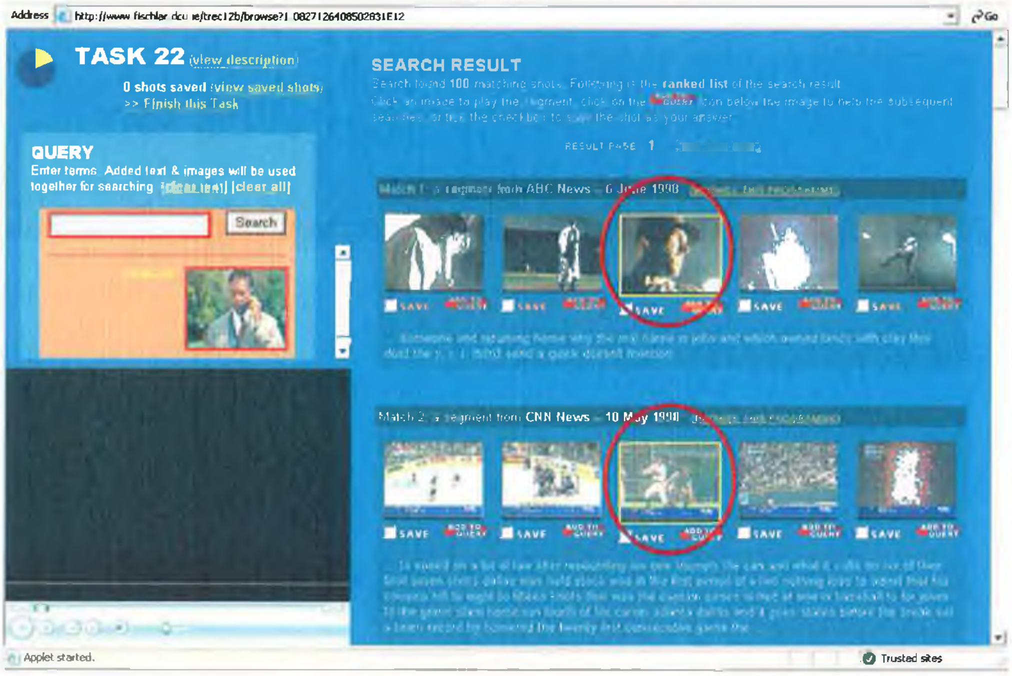
Topic Num	**Topic Examples**		
10			
22			

Figure 1-10: Unfriendly Image Topic examples from TREC 2003

As we can see from Figure 1-10, the examples from the topic 10 are all publicity material. Content from C_SPAN or TV news is unlikely to have matching visual content being more likely to contain outside footage of the Congressman being interviewer or footage of he as one of a group of people at a meeting or gathering. The second example of topic 22 has 2 stills taken from a movie and another promotional photo and once again it is unlikely that matching video will be found. However, there is a strong possibility of finding trees and greenery as the 2 example results for topic 22 shown in Figure 1-11 illustrate.


Address
http://www.fischlar.dcu.ie/trec12b/browse?1 0827126408502831E12
Go
TASK 22 (view description)
QUERY
Search
SEARCH RESULT
RESULT PAGE 1
SAVE
Applet started.
Trusted sites

0 shots saved (view saved shots)
>> Finish this Task

Enter terms. Added text & images will be used together for searching [clear text] [clear all]

Search found 100 matching shots. Following is the ranked list of the search result.
Click an image to play the segment, click on the QUERY icon below the image to help the subsequent searches, or tick the checkbox to save the shot as your answer.

Match 1: a segment from ABC News – 6 June 1998

Match 2: a segment from CNN News – 10 May 1998

Figure 1-11: Screen shot of the DCU TRECVID 2003 System

The screenshot in Figure 1-11 shows an image only search for "*Actor Morgan Freeman*" using one of the topic examples where low-level histogram-based features were used to generate the ranking. As we can see from the ranked results on the right (centre column) baseball features quite highly and the greenish background from the topic example is responsible for this. Note also that the ranked results also feature people.

1.7.5 The Future of TRECVID Video Evaluation

TRECVID 2004 continues with evaluation of television news content using a larger corpus of television news and updated query topics. Further into the future might see other domains like television sports (soccer and tennis for example) and other genres being evaluated. The feature extraction task will continue to have additional concepts added and removed depending on the corpus domain under evaluation, and possible additional features that could be included are:

1. River	6. Beach
2. Sky	7. Sunset/ Sunrise

3. Boats
4. Ski Detection
5. Music Genre

8. Football sports Field detection
9. Swimming detection
10. Laugh detection

1.8 Video Retrieval: Selected Areas of Active Research

1.8.1 Shot Boundary Detection

Shot boundary detection is one of the few research problems in the area of video segmentation that can be considered nearly solved, and performance of 90% is not uncommon for many systems [Browne et al, 2000]. While detection of hard cuts that make up the majority of shot transitions is fairly straightforward it is detection of the more gradual transitions like fades or dissolves where much of the current research work continues.

Despite the success of SBD, much improvement is needed in detection of the higher level semantic unit ‘scenes’. Most of our current success has been in the detection of scenes from specific domains like television news as we saw in TRECVID 2003. It looks likely that domain-centric approaches will become more active over the next few years.

1.8.2 Domain-Specific Object Segmentation

This area of Video IR research is currently focused on concept-based identification and extraction of objects like cars, buses, planes, buildings, boats, etc. Naturally detection of all object types across domain-wide content remains computationally infeasible and there has been some limited success in detection of these object concepts on limited domains [O Connor et al, 2003] [Browne et al, 2004].

Some object extraction approaches follow a semi-automatic approach, and what this means in practice is that they are assisted by a human who draws a rough boundary around the object of interest and the systems detect and track the object temporally through the video content [Ching-Yung Lin et al, 2003]. These systems could prove useful as an aid for annotators reducing their workload by reducing the frames of video they need to index.

1.8.3 News Story Segmentation

Research on television news segmentation has been ongoing for a number of years with the first large-scale system evaluation completed in TRECVID 2003 (see Section 1.5.4). Research in this area continues with work on story boundary detection, geo-temporal referencing and story personalisation. Retrieval performance on news specific content shows some promising results, due to the structured nature of television news story detection is possible.

1.8.4 Mobile Video Interaction

With mobile phones and PDAs capable of limited browse and playback of digital video there is a new area of possible video retrieval research. One fairly recent research aspect, driven in part by mobile operator's requirements for additional revenue streams to maximise return on their costly 3G licences, has been video event detection. The idea behind this research is that important action (goals, penalties, etc.) from live sporting events like football or rugby could be delivered to individual's mobile phones/ PDA's [Lee et al., 2001], [Lee et al., 2003].

Due to the nature of interaction with mobile video, research is continuing on navigation and retrieval on these devices. The size of the screen and keys makes interaction difficult, with high network bandwidth costs it is an important aspect to get correct.

1.9 Summary

In this chapter we started with a discussion on the growth of computing, and the need for improved video indexing and retrieval as we access and store increasingly large amounts of video content. A big reason for the success of the Internet has been the high performance offered by text search engines like Google, and we can expect demand to generate a similar performance requirement for other types of media like video and audio.

Text retrieval performance is generally quite high due to that medium's explicit semantic nature, indexing and retrieval of text content can also be quite straightforward. Video and other types of media have proved much more problematic due to their nature where semantic information is not added during their creation like a web or text document for example. The semantic information contained in a video is implicitly extracted when viewed by a human but not specifically recorded.

We discussed a number of the difficulties with improving video retrieval, and as we have said, semantic information in video is not yet explicitly recorded and it is not feasible to expect a computer to act like a human. Semantic feature extraction for the audio-based aspect of video has had good success with many groups using computer-based speech to text systems to extract rich semantic information from the audio. Doing likewise for the visual aspect of video still remains a difficult challenge.

Evaluation is an important aspect in the creation of retrieval systems. Since 2001 the text retrieval conference (TREC) has run benchmarking and evaluation for video retrieval systems, called TRECVID. This has evaluated shot boundary detection, news-based story segmentation, visual & audio-based concept feature segmentation and browsing and retrieval (User and System-based).

Finally we discussed some active video IR research topics; domain specific video segmentation in areas like television news and sports which have overcome some of the limitations of visual feature extraction by taking advantage of the domains nature. Section 1.5.1 provided an example of a news retrieval-based system.

Object-based retrieval is another important area of research. Much of the current work in this area focuses on specific concept-based extraction (cars, buses, planes, horses etc) for specific domain types like television news for example and we shall see more of this later in this thesis.

In the next chapter we will discuss the main video feature extractors which are used in the creation of a searchable video index.

Chapter 2 Feature Extraction from Digital Video

2.1 Introduction

In the previous chapter we discussed video retrieval, exploring some of the challenges as well as mentioning a number of Video IR solutions. In Section 1.3.1 from the previous chapter we discussed the three main levels of video features: low-level, medium-level and high level detailing how they related to video retrieval. In this chapter we will discuss the automatic extraction of these features themselves, without which, retrieval would not be possible. Extracted features can be broken into three main groups: visual, temporal and audio-based.

The purpose of feature extraction is to aid in the creation of an index for digital video. When a video index is available retrieval is possible and how well the retrieval performs naturally depends on the quality of the index that has been created. As we discussed in the previous chapter, indexing digital video is not straightforward, as semantic information is not explicitly recorded and is difficult to extract.

The first task in indexing raw video is to structure the content in some form. Video is composed of the following main structural units: frames, shots and scenes (see Figure 2.1). A shot represents a fundamental unit for retrieval with accurate detection a basic prerequisite for video browsing and retrieval [Browne, 2001]. After shot boundary detection, the shots themselves can be grouped into semantically related units called scenes, however accurate scene detection remains an unattainable goal for most content domains with television news the main exception [Smeaton et al., 2004 B].

Figure 2.1 illustrates how a number of shots fit into the video structure of frames, shots and scenes. While the example shows a clear visual difference between scenes, in reality the differences are less apparent.

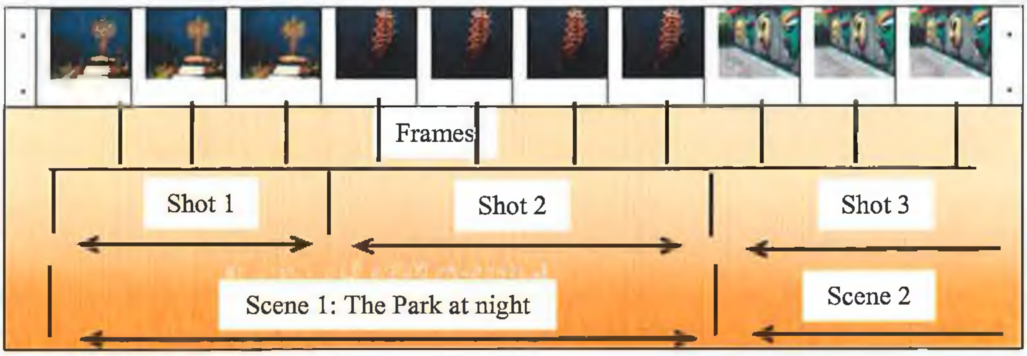


Figure 2-1: Visual Structure of Digital Video

The features that we will discuss in this chapter are usually aligned and retrieved at the shot level. As we have seen in the previous chapter most retrieval and evaluation systems use shots as their main unit for retrieval. This is not too surprising as scene-based detection is generally not feasible and the units are large.

2.2 Visual Feature Extraction

2.2.1 What is Colour?

Strictly speaking colour is actually light which at certain wavelengths are interpreted by the human brain as colour information. There are 3 types of pairs of receptors in the human eye and in each pair of receptors one is sensitive to low light levels while the other is sensitive to higher light levels. Each of the three pairs is sensitive to light at different wavelength frequencies, peaking at around 420 nm (a nanometre is one billionth of a meter) for blue, 535 nm for green and 560 nm for red. Figure 2-2 shows the wavelength and frequency of visible light.

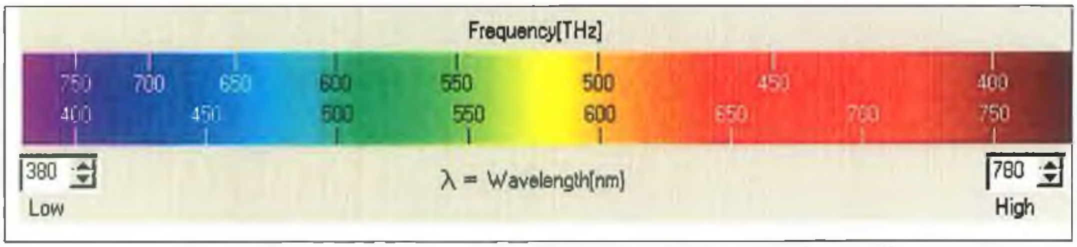


Figure 2-2: Visible Colour Spectrum taken from efg [efg, 2004]

There are also three types of colour perception that we are sensitive to. The first is the dominant colour/wavelength known as the hue. The second is the purity/ saturation of the colour, if there are other colours being detected by the eye at the same time the purity of a specific colour will be reduced (see Table 2-1). The third can be thought of as the surface luminance / brightness of the colour ([Earnshaw, 1985] page 1012)

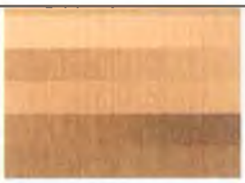
Table 2-1: Colour Saturation Examples

🟥 Red: Sat 100%	■ Red: Sat 20%	🟦 Blue: Sat 100%	🟦 Blue: Sat 20%

For computer-based image processing and display a square unit known as a 'pixel' is used to store the information about each individual colour and each image, picture or display will have a fixed number of such pixels. Typical resolutions for computer monitor displays are 800 * 600, 1024 * 768 and 1152 * 864. The larger the number of pixels available the greater the image size/ quality. A four megapixel digital camera records 2272 * 1704 pixels. A television will display an image with a resolution of 702 * 568 pixels.

Table 2-2 below shows an image of wood on the left and a close up view on the right using the zoom in option from a standard image processing software, notice the square pixels.

Table 2-2: Image Pixel example

 Image of Wood (1:1)	 Zoom-in (16X)

Colour Spaces

In order for colour to be recorded and displayed on a monitor/ television a colour model of human perception is used [Poynton, 2004] [ICC, 2004]. There are a large number of models in existence each with various proposed benefits and disadvantages, but it is worth noting that they are all just models of human perception and none are completely perfect. Each colour space can be used to generate a particular colour. The following are two of the most popular colour models available, i.e. RGB and YUV.

2.2.2 The RGB Colour Space

This is the most common colour space found on computer equipment, and one way to visualise this colour space is in the form of a 3D cube. Three colours RED, GREEN and BLUE are known as the primary colours and from these all other possible colours can be generated. These three RGB values, each integer value in the range 0 to 255 are used to generate the required colour. Computer monitors and television screens feature three electron beams representing the Red, Green and Blue frequencies that are used in the display of all colour information on screen. The RGB colour space is useful as it gives the strengths of each primary colour needed for visual display hardware like televisions and monitors. RGB is not however perceptually uniform as in some cases a small change in RGB values can have a larger perceptible visual effect. Table 2-3 illustrates a number of popular colours and their respective RGB values, as the table shows all colours RGB are made from degrees of red, green and blue.

Table 2-3: Colour and RGB Values

Colour	🟥	🟩	🟦	⬛	⬜	🟨	🟪	🟦
Red Value	255	0	0	0	255	255	255	0
Green Value	0	255	0	0	255	255	0	255
Blue Value	0	0	255	0	255	0	255	255
	Primary Colours					**Complementary Colours**		

2.2.3 The YUV Colour Space

This colour space has been with us for some time and separates the luma (Y) brightness and colour (UV) information. It was required in the early days of American broadcast television when a colour signal was needed that would be compatible with the more common and less expensive black and white television sets.

The colour information (UV) was ignored in black and white television sets but is viewable on colour sets. This colour space is still with us today in video compression standards like MPEG [MPEG]. The reason for this is that humans are more sensitive to brightness variations than colour fluctuations giving the algorithm a reason to compress the colour information more than brightness. A number of visual sources like images and television are encoded in YUV making it important for visual analysis.

The following is the standard formula used to convert RGB Values to YUV and vice-versa:

$$Y = R * \ .299 + G * \ .587 + B * \ .114;$$
$$U = R * -.169 + G * -.332 + B * \ .500 + 128.;$$
$$V = R * \ .500 + G * -.419 + B * -.0813 + 128.;$$

$$R = Y + (1.4075 * (V - 128));$$
$$G = Y - (0.3455 * (U - 128) - (0.7169 * (V - 128));$$
$$B = Y + (1.7790 * (U - 128);$$

2.2.4 Colour Feature Extraction

Digital Images stored on computer are made up of N1 * N2 square pixels, and the larger the value of N the greater the resolution/ quality of the image. Each pixel will have a specific colour assigned to it with the colour information encoded in a specific colour space. A popular digital image format JPEG (Joint Photographic Experts Group) can be encoded in a number of colour spaces, and RGB and YCbCr (YUV) are the most popular. The quality of standard television images is 702 * 576 pixels approximately. The features that are described in the following section operate on this colour information.

2.2.5 Colour Histogram Feature Extraction

The idea behind this is that colour values are divided into bands (known as bins) in a specific range and the larger the range the smaller the number of bins needed. Each pixel in the image or video frame is checked and when the relevant bin is found a counter for that bin is incremented. The collective set of values for the number of pixels in each bin gives an indication of the distribution of colour throughout the image.

There is no standard number of bins used in the generation of colour histograms and it can vary depending on the implementation. A larger number of bins mean that more space (in computer memory usually) is taken up with histogram information, fewer bins reduce the features' discrimination power so it is these that need to be balanced in any system [Browne, 2001] [Browne et al., 2000]. The following are a selection of popular histogram bin numbers 16, 32, 64 and 128, these are defined in the MPEG-7 standard [MPEG-7].

Colour histograms representing image or video frames can be compared in a variety of ways, and two of the most popular are absolute distance where the difference between bin values is totalled, the second is intersection where the bin intersection values are totalled.

The colour histogram has been used very effectively for shot boundary detection [Browne et al., 2000] however for large-scale visual retrieval it does suffer from poor discrimination power. Figure 2-3 is an example of two images that both generate **exactly** the same histogram illustrating one of the difficulties using the colour histogram feature.

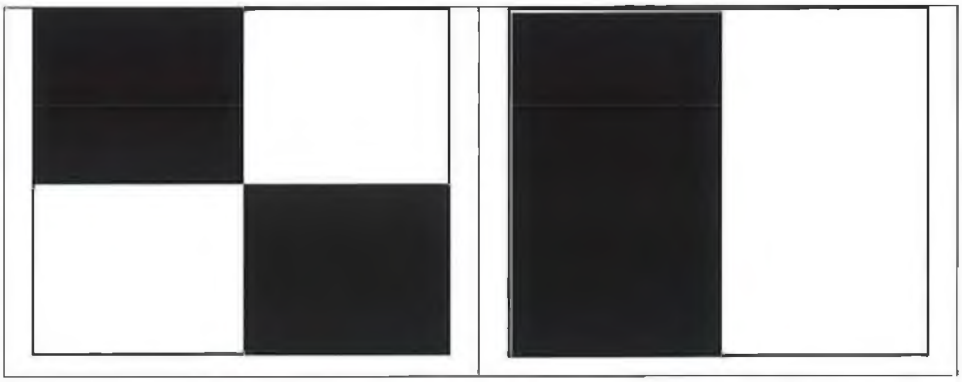


Figure 2-3: Similar Histogram Example

One enhancement to the histogram feature is that instead of operating on the global image a regional approach is used whereby the image is divided into a number of fixed regions with a histogram created for each one. Figure 2.4 shows three example images and their associated 4 region colour histograms. For the Buddha statue, we have four regions all with peak values for red. For the bridge over lake, two regions feature peak values for blue.

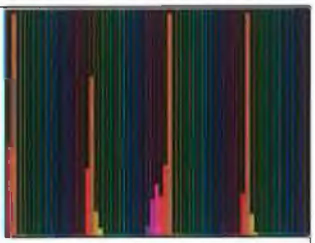
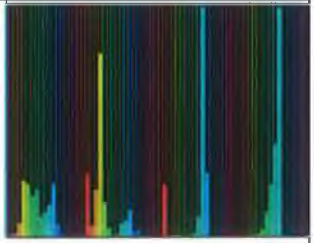
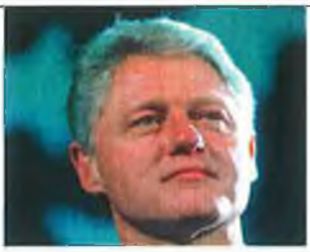
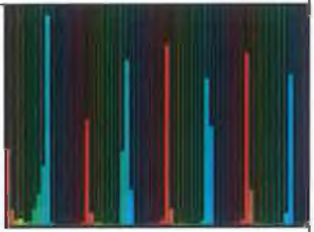
1	 Buddha Statue	
2	 Bridge over a Lake	
3	 Bill Clinton	

Figure 2-4: Colour Histograms

Colour histograms can be quite effective for query topics like "*Find me video of boats sailing on the sea*" as valid results will usually contain a large blue background or "*Find me video of trees in a forest*" as valid results will usually contain a large green background. The second image example in Figure 2-4 also illustrates an effective query as the background shows high levels of green and blue illustrated by its histogram to the right.

On the other hand query Topics like "*Find me video of ex American President Bill Clinton*" would prove more difficult to retrieve using colour histograms as valid results could feature many different backgrounds with each generating large histogram differences and consequently a low ranking. The colour information in the case of this query topic is not important.

2.2.6 Colour Average & Mode Feature Extraction (1st order Colour Moments)

This colour feature extractor works by looking at specific distributions of colour in the image. Usually a regional approach is employed on *N* (X * Y) fixed regions. The main values that are extracted from each region are the average and mode colour occurring in the region. The calculation is straightforward enough, for mode colour you simply calculate the histogram for the region and output the colour of the bin with the highest number of entries (for the RGB colour space we would need to output three values for each region R, G and B). Colour average is calculated by totalling each of the colour pixel values in the region and dividing by the total number of X * Y pixels in the region. The number and size of regions can vary depending on implementation with a standard approach dividing an image into 9 fixed regions. [Smith et al., 1996]

2.2.7 Colour Correlograms

One aspect that is missing from a colour histogram is the spatial distribution of the colours and as a result completely different images can generate similar histograms. The colour correlogram addresses this problem by including the colours spatial information with colour pairs providing a probability of a particular colour **C** at distance **D** in the image. The value of **D** is variable with a large value requiring longer processing and storage time. The idea behind this approach is that the colour values in the image are quantizied into colour ranges, each pixel in the image is checked and probabilities calculated for colours at a given distance D.

The correlogram is essentially a table of colour pairs which contains the probability of colour **Y** at distance **D** from colour **X**. In order to store the correlogram efficiently they choose a small value for **D** [Hung et al., 1997].

2.3 Edge Feature Extraction

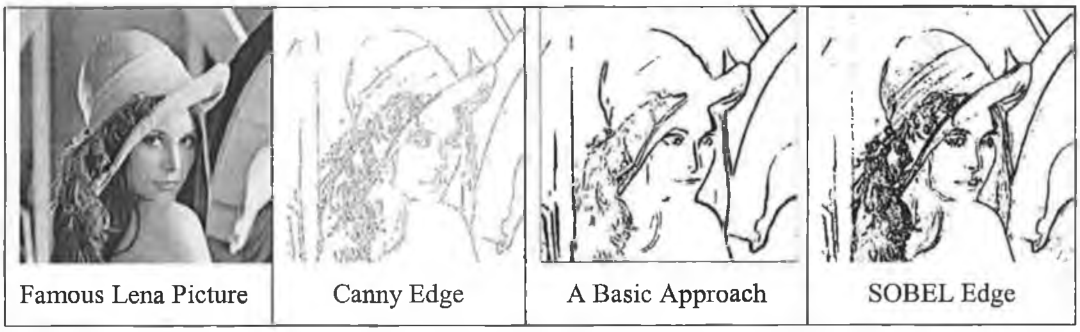
Edges can be thought of as boundaries or areas with large differences in texture/ colour. It is generally regarded in perceptual science that humans are especially sensitive to these boundaries and we use these in our object identification process ([Earnshaw, 1985] page 1015).

Research into edge detection has been ongoing since the early days of computer image manipulation and there are a number of algorithms for detection of edges in images with the most popular being SOBEL [Gonzalez et al., 1992] and Canny [Canny, 1986]. Research into improving the accuracy of edge detection is ongoing as it is vital for object detection and extraction.

A straightforward edge detection approach is to devise a difference threshold among neighbouring pixels over which a pixel is marked as an edge. Each pixel in the image is checked with each adjacent pixel and if it is over the threshold the pixel is marked as black (edge) otherwise it is marked white (pixel (X,Y) is compared with (X+1,Y) and then (X, Y+1)).

Edge detection can be tricky due to the nature of thresholding of pixel differences, visual lighting conditions or the image recording process. Many edges will fail to be detected completely or at all, as their differences might be too low to be passed by the threshold. Specific image and regions may need a higher or lower threshold in order to be detected correctly. As a result an edge-detected image will contain some broken and missing edges.

Figure 2-5 illustrates the output from three edge detection methods on the popular Lena image using standard thresholds. Note how some edges are missing or broken particularly the top of the hat which no method detected correctly.



Famous Lena Picture | Canny Edge | A Basic Approach | SOBEL Edge

Figure 2-5: Edge Detection Variations

2.3.1 Edge Histogram Extraction

The edge histogram feature is similar in operation to the colour histogram hence the name. It works by defining a number of pre-defined edge pixel orientation layouts. The pixels from detected edges in the image are checked for matches to these predefined layouts and a count is incremented when a match is found. The layouts/ patterns are devised to find sloping, horizontal and vertical lines mainly with the idea that visually similar images will contain approximately equal numbers of these patterns [Park et al., 2000].

Figure 2-6 illustrates 16 possible edge shapes (histogram bins) that could be searched for in an image. The blue indicates an edge pixel while red indicates that we expect to find no edge pixel. White indicates that we don't care if there is an edge pixel or not.

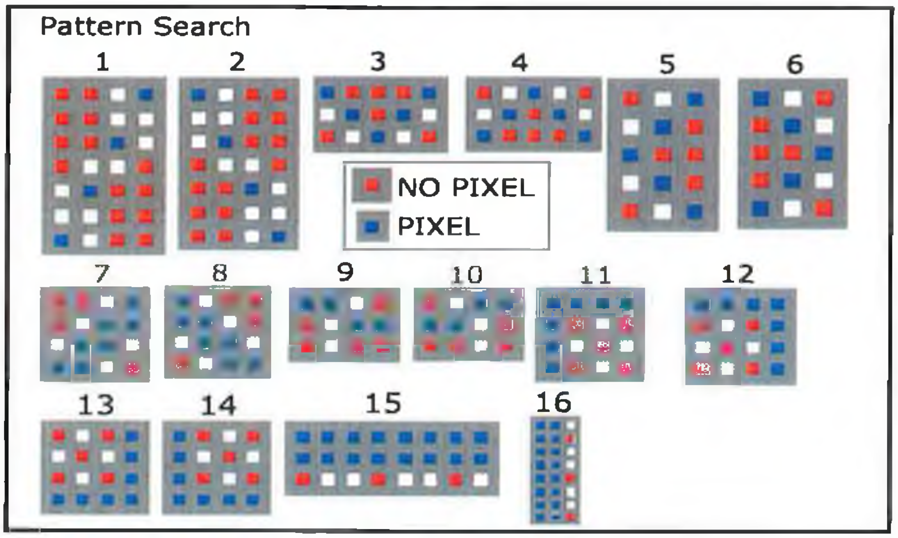


Figure 2-6: Edge Histogram Shapes

The approach can be improved by segmenting the image into regions before calculating the edge histogram, however there is a trade-off between improved retrieval performance and index size. An edge histogram approach gives a similarity comparison on the quantity of various sloping lines and thus can still lead to images with very different shapes generating similar edge histograms.

2.3.2 Texture & Coarseness Detection

Visually an image can be composed of many different types of textures from a grass field to a person's plain-coloured shirt. The examples of edge detection we have shown earlier remove most of the textures due to a high difference threshold, though this can be changed by running with a lower threshold. Figure 2-7 illustrates some of the different textures possible:

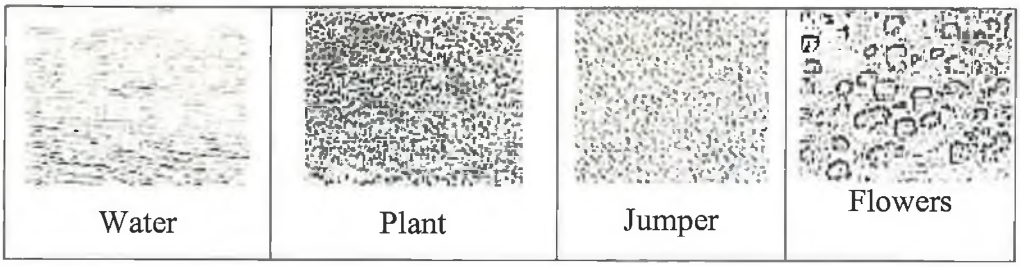


Figure 2-7: Various Image Textures

Detection of these textures can provide an additional method for search or can be used to supplement concept-based features like water detection. Indexing differences in texture can be used in combination with other features to detect more specific semantic scenes and objects, an example might be a feature detector that detects images containing beaches, colours and edge texture could identify water and sand.

As we can see from the texture of water from Figure 2-7 there is a large difference in this texture compared to the other examples. If enough examples were obtained it could provide an input for water detection, and this texture feature is useful as not all water is blue especially under dark and low light conditions and colour alone is not enough to identify it.

2.4 Object-based Feature Extraction

Object-based feature extraction, which as we will see is an important part of this work, is primarily based on edge information from the image or video frame. One of the main reasons for this is that specific colours can vary greatly under different lighting conditions making object detection a haphazard affair, and while colour information is useful for object detection it is extracted separately. Object-based detection and extraction on natural video is one of the most challenging areas for feature detectors and has so far only had limited success. The following requirements for object detection explain why this is the case:

[1] Invariant to scale: The object could fill the full image or be in a small area. This is very difficult when the image contains many other types of object information, the background is usually noisy, and the edge detection itself is far from perfect.

[2] Invariant to rotation: A matching object in the image might be rotated along the X, Y or Z-axis and the algorithm needs to take account of this during detection.

[3] Occlusion: This is where part of the object is hidden from view either by another object or the object itself. This is another big challenge as the algorithm needs to be able to fill in the missing data and in many cases there could be more than one possibility. Morphological filtering can be used here to remove some edges to make detection easier.

[4] Variation: There are many different variations of the same object type which generate very different comparison scores. Take a car detector for example: if we picture all the different types of cars we can think of it is easy to see how different object matches could be produced. Modelling detection of cars would have a tough time handling this and that is before we even start to worry about the other problems above.

[5] Noise: These are edges from other objects in the image and/or errors in the edge detection process itself. Processes like morphological filtering can fill edges in and reduce others but there are limitations.

[6] Lack of 3D Information: Images and video are a 2 dimensional representation of a 3D world, depth information is not recorded. For a completely symmetrical object it does not matter what angle we look at it from as the shape will always be the same, like a vase, but most objects are not like this.

2.4.1 Fully Automatic Object Detection Systems

In light of all the difficulties described above it is generally considered that detection of all possible object types is generally considered a computationally infeasible problem. While this is certainly the case (currently) there has been some success with limited object-based classifiers, and these are available for a selection of specific concept-based objects like chairs, planes and cars [Amir et al., 2003] [CDVP].

These classifiers are trained on examples of the particular object from a subset of an evaluation corpus (see Section 1.5 TREC) and over time learn to detect the object. One problem with this approach could be the following: imagine we are training a horse detector with hundreds of pictures of horses in a green field as our valid training images. Now we pass it an image of a policeman on a horse in New York City next to a busy street, it is easy to see a possible detection problem with an image so different to the training set. Imagine instead that we test it with a couple of animals in a zoo, these could also be similar to a horse and detected as valid. A possible solution to this is to use invalid examples in training. No training can provide a wide enough variety of all valid and non-valid examples to remove all error. Concept-based detection performance has given mixed performance as we can see from Figure 1.6 and 1.7 in Chapter 1, and construction of concepts for all object types is non feasible. However, some work is progressing on the creation of ontological-based concepts for genres like television news and sports.

2.4.2 User Assisted Object Segmentation

Another area where work has been done on object detection is via user assisted segmentation, which is particularly useful in real video content. The idea behind this concept is that the user aids the segmentation process by roughly sketching around the boundary of the object and tagging it and, the system can then accurately detect and track this object through the content. An example of this process can be seen in the QIMERA system (see Section 2.4.4).

A number of participating TREC groups led by IBM completed a manual annotation project on the TREC 2003 development content [Lin et al., 2003]. Each of the 20 participating groups was given a specific number of videos to annotate using an annotation application designed by IBM (see Figure 2-8). Each shot in the content was annotated by clicking checkboxes if the shot contained any of a set of 130 pre-defined features, and after clicking okay the user draws a box around the location of any identified objects. The idea behind the project was that groups shared the burden of annotation but had access to the complete annotated index after completion.

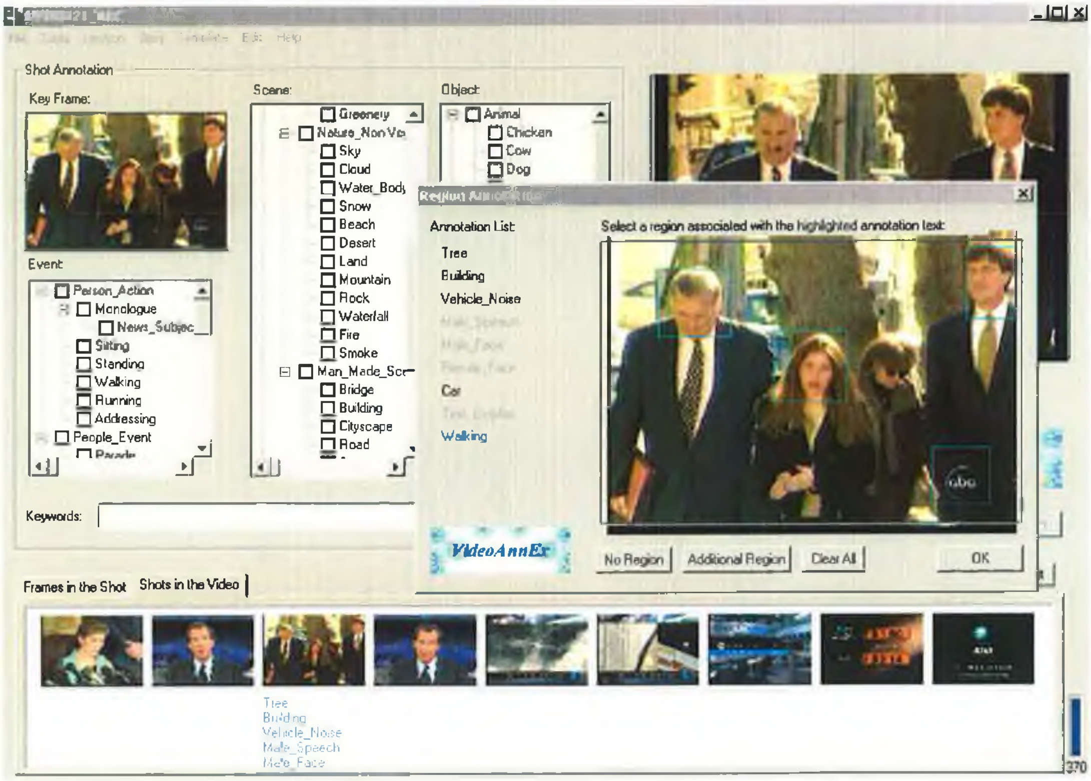


Figure 2-8: Screenshot from IBM Tool for TRECVID Collaborative Annotation 2003

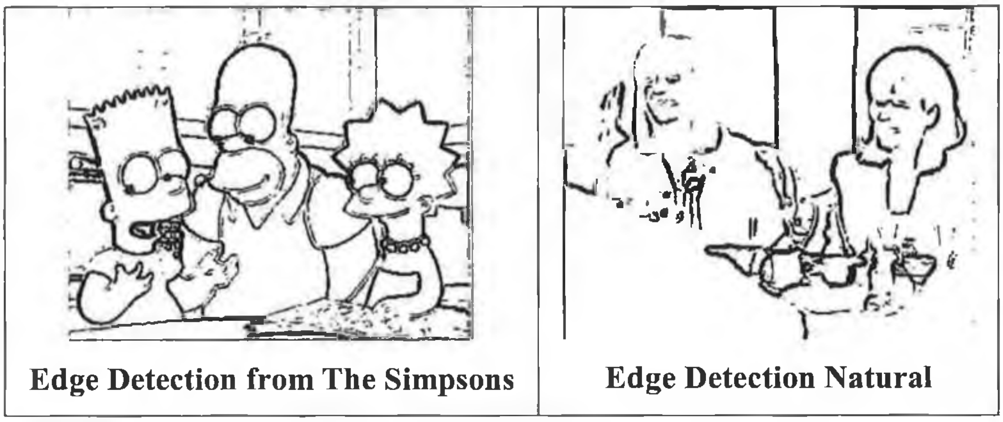
2.4.3 Object Template Matching

Another object segmentation technique that can be employed is the use of object templates. The idea behind this option is that a number of examples (templates) of the object to be detected are extracted by the user in various poses. A shape-matching process is used during the comparison/ detection to compare each image against the available templates [Adamek et al., 2003]. The template matching method performs better on animated content compared to natural video for three important reasons:

- In animated content such as cartoons, the shape and colour of the characters and backgrounds do not tend to differ greatly throughout an episode or an entire collection, for natural video people change facial expressions constantly and colours fluctuate under the variable lighting conditions.

- In animated content, characters tend to have a limited number of detectable and repeatable expressions, whereas in natural video an individual's expressions tend to be more subtle and varied.

- In animated content the boundaries for each of the objects are very clear and detectable because a reduced set of colours compared to natural video is normally used whereas, the edges in natural video are more broken and separated. To illustrate, Figure 2-9 shows edges automatically detected from an animated image (left image) and from a natural scene (right image).


Edge Detection from The Simpsons

Edge Detection Natural

Figure 2-9: Animated and Natural Image Edge Detection

Retrieval has been successfully applied on Simpson's animated content on objects extracted via template matching [Browne and Smeaton, 2004]. The system demonstrated retrieval on 20 hours of The Simpsons content with object-based search available for 10 of the main characters. Object information was supplied by [Adamek et al., 2003] who used a combination of homogenous yellow templates of the main characters and a shape matching algorithm to extract object information. We will discuss this system in more detail in Chapters 4 & 5.

2.4.4 Object Extraction from Natural Video

As we have discussed, object extraction from natural video presents a real challenge. A number of systems have been developed that attempt to identify objects by their active movement. The idea behind this concept is that the foreground objects will move from frame to frame to a greater degree than the background which will exhibit less movement. The QIMERA system is a collaborative video segmentation platform that incorporates natural object detection using a number of colour and texture features to identify object boundaries. They also demonstrate user assisted object segmentation on a number of standard MPEG test sequences [O'Connor et al., 2003] see Figure 2-10.

Figure 2-10: MPEG Table Tennis Sequence (Taken from QIMERA [O'Connor et al., 2003]).

2.5 Video Text OCR

Video text OCR is another feature worthy of mention. The idea behind this feature is that text displayed on screen is converted into machine readable ASCII text via an optical character recognition process (OCR). A variant of this feature is used to convert image scans of books and documents into machine-readable text.

Television news programmes frequently feature banner text giving vital semantic information that may not be supplied by the spoken audio. Figure 2-11 shows an example of the text OCR process, where edge detection is used to identify boundaries, then it is checked for groups of vertical and horizontal edges (hopefully text). When suitable groups are identified the application attempts to identify the text-based on a machine learning approach.

This process is difficult as the text might be overlaid onto an image with areas of matching colour, if this is the case the edges of the text might not be detected correctly leading to missing or unidentifiable characters. Figure 2-11 is an example of the video OCR process:

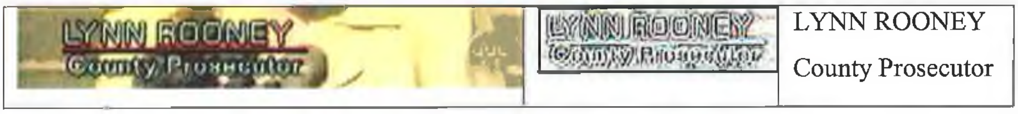


Figure 2-11: Video Text OCR Example

2.6 Temporal Feature Extraction

Temporal features are based on the temporal nature of video content and use visual and/or audio layer changes over time to extract features. There are four specific types of temporal feature which we will discuss in this section with the first (shot boundary detection) being perhaps the most popular and successful.

2.6.1 Shot Boundary Detection

As we discussed in Section 1.4 shots are "continuously recorded images captured by a single camera source" with transitions between shots being a good example of detectable temporal events. Shots are part of the video structure and a minimum unit for browse and retrieval of digital video making their accurate detection essential.

Shot cut detection is normally based on adjacent image comparison, and when a large enough image difference is detected between shots, a shot cut is registered. Colour histograms are an example of one feature that can be used in the comparison process. Detection of other types of shot transitions such as dissolves require a more complicated procedure as the image change is more gradual, occurring over a number images and may not be detected by a specific threshold for adjacent frame differences. Rules such as a sliding window of N combined images can aid in their detection [Yu-Fei et al., 2001] [Volkmer et al., 2003].

2.6.2 Video Motion Extraction

This type of temporal feature extraction tries to extract the motion of the recording camera. There are a variety of different camera motions possible and they are important as they can give us an idea of the activity and importance of the content. The following are some of the main camera motions:

Zoom-in: The camera gradually focuses in on a distant object or scene from a wide view.

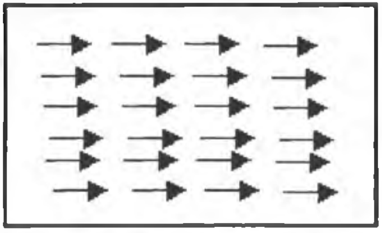
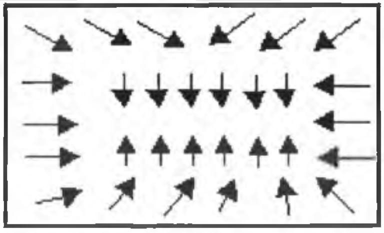
Zoom-out: Basically the opposite of a Zoom-in, the camera starts focused in on a particular object or scene and gradually focuses out to a wide scene.

Pan: Instead of recording in the same position the camera is rotated on the horizontal axis. We can see this in action when viewing landscape content were the camera is rotated left to right to record all the scenery.

In a camera pan for example, the shot tends to be an establishing shot and is used to give people a better view of a location. Nature and holiday programmes use camera pans quite regularly. A zoom-in is used to focus on a specific object of importance, for example a Formula-1 car taking a high-speed corner. A zoom-out is used to show a wider field of view, for example several Formula 1 cars racing for position. Video motion can be used in combination with other features in the detection of sporting goals and other events [Sadlier et al., 2003].

Motion is extracted by monitoring the movement of object edges and colours over time and plotting their moment, and Table 2-4 gives an example of this expected movement and resulting motion. Calculation of this movement is actually performed by a number of video compression technologies such as MPEG-1 and this means that calculation can be done quickly on the compressed video itself [Tan et al., 2000] [Donnelly et al., 2003].

Table 2-4: Example of Camera Motions

 **Camera Motion: Pan Left**	 **Camera Motion: Zoom In**

2.6.3 Temporal Correlograms

The temporal correllogram feature is really an expanded colour correllogram that not only includes the spatial distribution of colours in a specific image frame but also over time. Correllograms are computed for all **N** frames of a video segment (usually a shot) with the colour probabilities stored for all frames. The temporal correllogram gives the probability of colour **X** at distance **D** from colour **Y** in frame **L**.

MediaTeam in Finland [Rautianen et al., 2002] used this approach on 11 hours of TREC 2001 video content with images indexed in the HSV colour space, and their initial results were promising but need evaluation on later TREC's before a clear picture of performance can be obtained.

Attempts by the group on TREC 2003 [Rautianen et al., 2003] showed results slightly above the median. This is to be expected, as the feature is still a low-level feature extracting limited semantic information.

2.7 Audio Feature Extraction

2.7.1 Importance of Audio

Audio information contains an array of important features, words in the form of human speech, music and sound effects. The area of automatic speech recognition has been under intensive research since the 1960s and can now be found in many automatic typing tools such as IBM's ViaVoice. Applications such as ViaVoice are controlled by a human voice with the user talking to the computer using a microphone. The generated audio is analysed by the application with matching words displayed on screen. A period of training is generally required by these applications in order to learn the vocal traits of the individual before they can be successfully used to type documents.

2.7.2 Speech to Text extraction

Speech to text is currently the most important semantic feature that we are able to extract automatically from video and is the only real high-level feature that we have available. Automatic Speech Recognition (ASR) has been very successfully applied to video content with word error rates of approximately 30 percent [Gauvain et al., 2002]. This feature has been one of the main TREC Video evaluation features since it started in 2001. Even high word error rates can generate good search results for users.

Table 2-5 illustrates the quality of ASR from a random TREC Video Shot. Looking at this text it is easy to see the potential retrieval power of ASR text. While the detection is far from a perfect we can get an idea of what is being discussed from reading the text, important keywords like "steve fossett", "balloon" and "solo spirit" are detected correctly. It's worth pointing out that this performance is not always the case as other noises and music playing on the audio can reduce the quality somewhat.

Table 2-5: Example of LIMSI ASR text and Actual Speech from a TREC 2003 shot

ASR	steve fossett and his balloon solo spirit arsenide over the black sea drifting slowly towards the coast of the caucuses his team plans if necessary to bring him down after daylight tomorrow in the chechen capital of grozny unable to favorable currency the jet stream to that espy malfunctioning and leaders and other assisted with his second attempt nathan under the weight despite last you did short of his goal to be the first balloonists to fly around the world nonstop sheila macvicar a.b.c. news london
Actual	steve fossett and his balloon, "solo spirit," are tonight over the black sea, drifting slowly toward the coast of the chechnya. his team plans, if necessary, to bring him down after daylight tomorrow near the chechen capital of grozny. unable to reach the favorable currents of the jet stream to the south, cursed by malfunctioning heaters and other systems, this second attempt may flounder the way his flight last year did, short of his goal to be the first balloonist to fly around the world nonstop. sheila macvicar, abc news, london.

As we discussed in detail in the earlier chapter, document 'text' offers rich semantic information on which to perform retrieval. It is worth noting at this point however that ASR does differ from document text. The following are some of the main differences:

- In many cases what is said on screen will not always reflect what is visible on-screen. Dialogues between characters for example will generally not talk about visual information such as their location, their names, what they are doing and what objects are visible in the background. In television news broadcasts, some of this information is displayed on-screen instead of spoken.

- Spoken audio may discuss visual content that has been just seen, but the visual content may not correlate with what can be heard. ASR-based retrieval that ranks shots may show the user keyframes from invalid shots.

- Word error rates. No ASR system is perfect and an error rate of 30 percent is actually considered quite good. This can be increased still further when music and other non-speech noise is quite loud. However, good retrieval performance can still be maintained even though the word error rate is high [Barras et al, 2002].

2.7.3 Speech Music Extraction

Speech-Music determination is an example of a concept-based audio feature that provides a confidence level of audio containing speech or music, and is usually aligned at the shot level. The audio stream is analysed for speech and music signals matching recognised patterns, the algorithm outputs a confidence measure for each (usually aligned to the shot level) [Jarina et al., 2002]. This feature has been used for TREC evaluation for the last number of years.

2.7.4 Loudness and Voice Pitch

The vocal pitch and loudness of the audio are another two important low-level audio features. The audio is especially useful for video sports summary and event detection. With this type of content one would expect a television commentator's voice to become quite high in the event of a goal scored or some other important event. Recorded crowd noises would also become notability louder [Sadlier et al., 2003].

2.8 Summary

In this chapter we focused on feature extraction from digital video. The aim of feature extraction is to obtain semantic information from the content, and this is not a straightforward task as semantic information in digital video is not explicitly recorded as it would be in the case of a text document.

Feature extraction can be broken into three main areas, visually-based, audio-based and temporally-based. Semantic information extracted by these features is at three main levels (low, medium and high). There is a large gap in the level of semantic extraction for medium to high features compared to that of low to medium features. High level features offer the best retrieval performance of which text extracted from audio ASR is our only real example.

Visual features can be broken into colour, edge and object-based. The most well known example of a colour feature is the colour histogram that stores the quantities of colour ranges and uses these in the comparison process. Edges are the boundaries between objects and are used in the creation of the edge histogram feature and object detection. We discussed two main methods of edge detection SOBEL and CANNEY and explained that the process of detection is not perfect with error unavoidable.

A number of methods for object extraction are discussed from user assisted to template shape matching. There are five main reasons given why the detection of objects has proved to be such a difficult challenge: invariant to scale and rotation, occlusion, variation and noise.

Temporal or time-based feature extraction maybe applied to visual or audio information with shot boundary detection being the most common example. Other temporal features that were discussed were camera motion detection and temporal correllograms.

The final feature type we discussed is audio-based. The most powerful feature that we can currently extract from video automatically is text extracted from the spoken audio. This feature is the only high level feature that we have for video and is very important for retrieval. There are a number of differences between document text and ASR that make extraction of data from the visual layer important. Document text is descriptive and informative, while ASR text contains information from spoken audio but implicit information such as the individual's location, names and what objects are visible in the background is very often missing.

In the next chapter we will discuss information retrieval in more detail and detail current video retrieval solutions.

Chapter 3 Information Retrieval and Relevance Feedback for Video Retrieval

3.1 Introduction

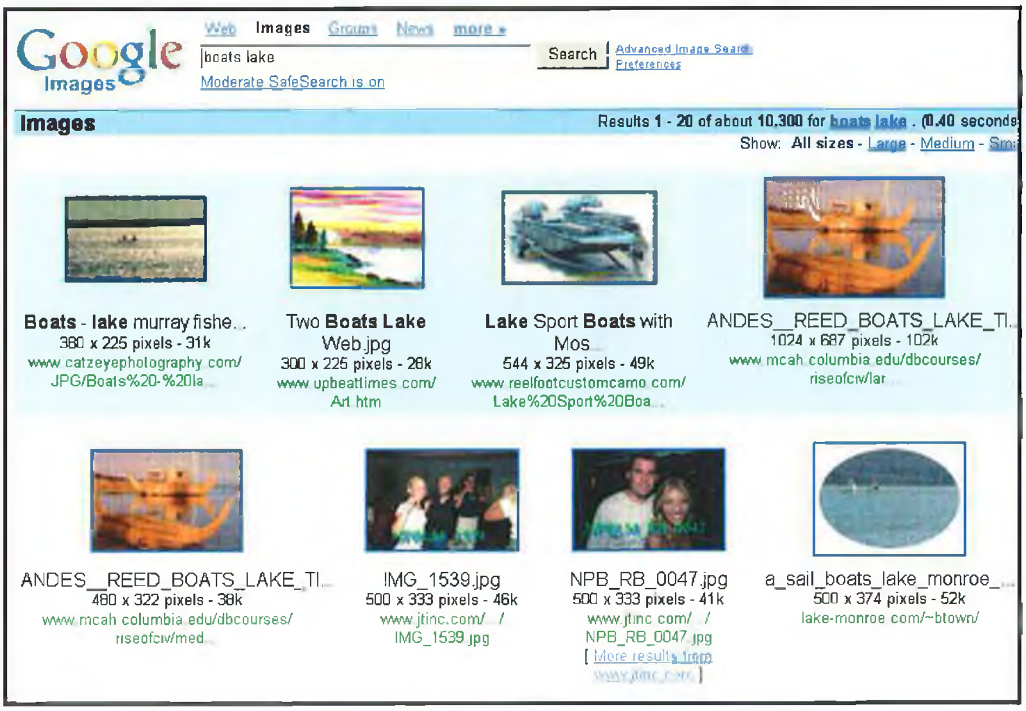
In the first chapter of this thesis an overview of digital video retrieval was given with discussion of some of the main challenges and solutions and the second chapter focused on various methods of video-based feature extraction. In this chapter we will discuss Information Retrieval in more detail and focus on a specific model of retrieval, that is Relevance Feedback.

The future will almost undoubtedly see an increase in the need for more effective retrieval of non-text media like digital images and video. Until the early 1990's computers were very restricted in the quantities of non-text media they could store, but with the advent of cheap storage and efficient media compression this is certainly not the case today.

The field of Information Retrieval (IR) is concerned with returning to the user information that they require, accurately and in a timely fashion. We can see a good example of this with today's web search engines that return ranked lists of web pages from all over the world. While search engines primarily return text pages this is by no means the only type of media for which they facilitate retrieval. In recent years options like image search have also been made available by large search engines.

Current web search engine approaches to image retrieval like GOOGLE's (Figure 3-1) do not index the media directly but instead make use of the structure and text from web documents themselves. The indexing approach uses the HTML tags around an embedded image, surrounding text and the hypertext links from the document to return a ranked list of images to the user. The idea behind this approach is that the image is related to the text from the web document that includes it. As earlier chapters have shown, associated text contains a great deal more semantic information than we are currently able to extract automatically from images and this makes the approach quite efficient and accurate.

The image below demonstrates a Google image search with the keywords “boats” and “lake”. Each of the results is linked to an image that contains the matching keywords in the image file names or in the case of the 6th and 7th result the URLs actually contain a keyword match.



boats lake

Moderate SafeSearch is on

Results 1 - 20 of about 10,300 for **boats lake** . (0.40 seconds

Show: **All sizes** - Large - Medium - Sm

Boats - **lake** murray fishe...
380 x 225 pixels - 31k
www.catzeyephotography.com/
JPG/Boats%20-%20la...

Two **Boats Lake** Web.jpg
300 x 225 pixels - 28k
www.upbeattimes.com/
Art.htm

Lake Sport **Boats** with Mos...
544 x 325 pixels - 49k
www.reelfootcustomcamo.com/
Lake%20Sport%20Boa...

ANDES__REED_BOATS_LAKE_TI...
1024 x 687 pixels - 102k
www.mcah.columbia.edu/dbcourses/
riseofciv/lar...

ANDES__REED_BOATS_LAKE_TI...
480 x 322 pixels - 38k
www.mcah.columbia.edu/dbcourses/
riseofciv/med...

IMG_1539.jpg
500 x 333 pixels - 46k
www.jtinc.com/.../
IMG_1539.jpg

NPB_RB_0047.jpg
500 x 333 pixels - 41k
www.jtinc.com/.../
NPB_RB_0047.jpg
[More results from www.jtinc.com]

a_sail_boats_lake_monroe_...
500 x 374 pixels - 52k
lake-monroe.com/~btown/

Figure 3-1: Results from GOOGLE Image Search

3.1.1 Overview of Traditional Video IR

Most text-based IR models of retrieval currently in use involve the choice of a number of keywords by the user describing their information need and, as seen from the previous section, this same approach is followed for web-based text and image search. Video-based media offers more numerous methods of search that can be employed by the user, and these can be used both individually and combined, during retrieval. The following are five different video IR search methods based on available video features.

[1] Standard Browse & Playback: This is the most straightforward approach to video IR whereby a list of video programmes is displayed to the user. If the list contains too much content the programmes can be sub grouped into genres and categories (Figure 3-2). The user searches the list and selects a video of interest and usually the name of the video programme is somewhat descriptive. Having made a selection, the user then browses through the video shots before selecting the segment of interest and commencing playback. This approach works well if the video content is known to the user and the library is not too excessively large. However a large and unknown corpus of video like that supplied by TRECVID 2003 would provide a real challenge to this approach [McDonald el al., 2001] [CDVP, 2004].

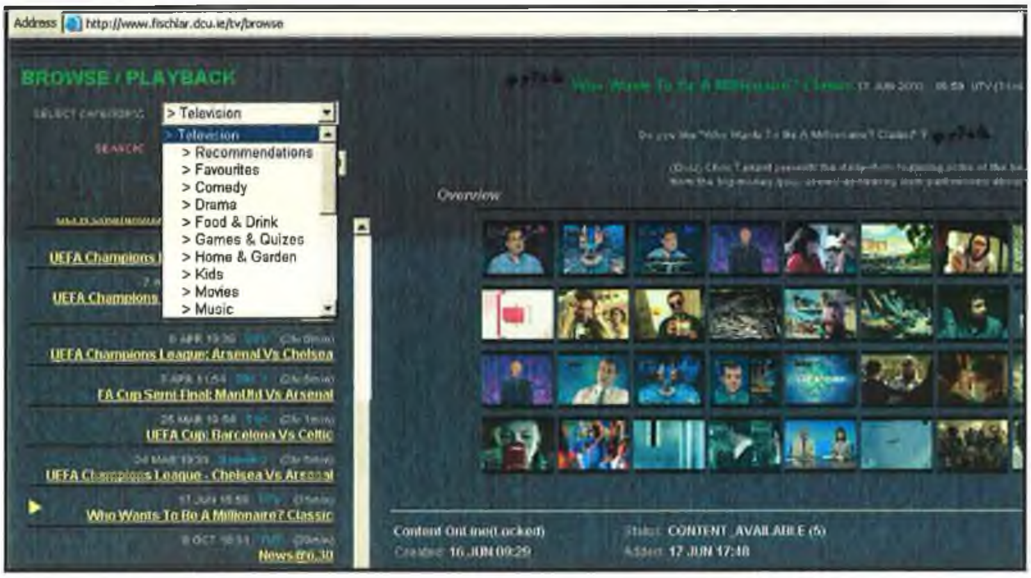


Figure 3-2: Físchlár TV Screen

Figure 3-2 illustrates a standard browse & playback interface. The recorded programme list is displayed on the left with programme title, length, broadcasting time, date and channel, and the programmes can be displayed in various categories as shown such as Drama, Comedy and Kids. Shots from a chosen programme are shown on the right with a brief text description and a group of selected images as the summary of the programme. Video playback is provided to users by clicking on any of the displayed image icons. [Smeaton et al., 2004]

[2] Text-based Search: The user inputs text keywords describing their information need in a similar fashion to a traditional text-based search. This offers a big improvement over the previous method as text is semantically rich giving a very powerful means of retrieval. The supplied text is extracted from the video's audio layer via an ASR approach or extracted from the closed captions (subtitles) although the latter does suffer from a shot alignment issue.

Before extracted text can be used for retrieval it will need to have standard information retrieval techniques applied to it like stemming and stop-word removal. Stopword removal eliminates commonly occurring words 'like', 'and', 'or', 'the', 'I', 'me' etc. from being indexed, as these words are not effective for retrieval because they are so common and non-discriminating. Stemming is a process whereby each variation of a word's morphology is reduced to a single word stem. So "stem", "stemmed", "stemming", "stemmers", and "stems" are all reduced to the one form, namely "stem" using a relatively straightforward set of rules. When the initial processing of text is completed the remaining terms are indexed, and as with most video features the processed text is generally aligned to video at the shot level.

A major issue with video text is that it is extracted from the spoken audio which can be missing implicit information like character names, background objects, places and occurring events. These aspects are not always discussed or mentioned by characters as part of the dialogue and as a result text-based searches can fail.

Video IR systems using this approach generally have their supplied text aligned at the shot level with ranking achieved via a traditional text IR approach like tf * idf (Equation 3-1) or BM25. The general assumption of these approaches is that less frequently occurring terms should be weighted higher than more frequently terms.

Before the text can be indexed terms (words) are stemmed. This reduces words like "reading" to "read" and "happens" to "happen". After this is completed, frequently occurring words like 'the', 'and', 'I', 'he' and 'will' (stopwords) are removed.

In text retrieval a popular term weighting method uses term frequency (**tf**) by the inverse document frequency (**idf**). Using this approach terms are weighted differently based on their discriminating power. Terms that occur more frequently are given a reduced weight in comparison with less frequently occurring terms. The formula weights each term based on the number of documents in the collection that have the term, the number of terms in each document and the overall number.

$$W_{ij}=tf_{ij}*\underline{\log_{2}\frac{N}{n}} \qquad \text{(Equation 3-1)}$$

$$\xleftarrow{\qquad} \boldsymbol{idf}$$

w_{ij}	weight of Term T_j in Document D_i
tf_{ij}	frequency of Term T_j in Document D_i
N	number of Documents in collection
n	number of Documents where term T_j occurs at least once

Another weighting method for the probabilistic model of text information retrieval is known as Okapi BM25 which considers document length in term weighting and is one of the more popular text IR approaches currently in use [Robertson et al, 1992].

[3] Query by Example: The user provides an image or video clip that matches or closely resembles their information need, video is ranked based on visual similarity to this content and a ranked list of results is returned to the user. This approach is a popular method of starting an initial query search or in relevance feedback based retrieval. We will discuss the computation of this in more detail in Chapter 4.

Retrieval performance depends on the quality of the example or query and perhaps unsurprisingly its similarity to the stored content. TRECVID provides a number of still image and video examples for each of the query topics they generate, and many participating groups incorporate query-by-example methods in their video IR systems [Browne el al., 2003].

[4] User Sketch: This option is similar to the previous approach except that instead of providing an existing image, the user draws a visual representation of their information need which can be in the form of a freehand drawing and/or a colour

image representation. This 'image' is usually broken down into various low level features and compared against the video content with results returned based on their similarity. Performance with this approach can be limited as retrieval is generally dependent on low level features like colour and edge histograms, however if the query topic can be found by a search for specific background colours or shapes then good performance is achievable.

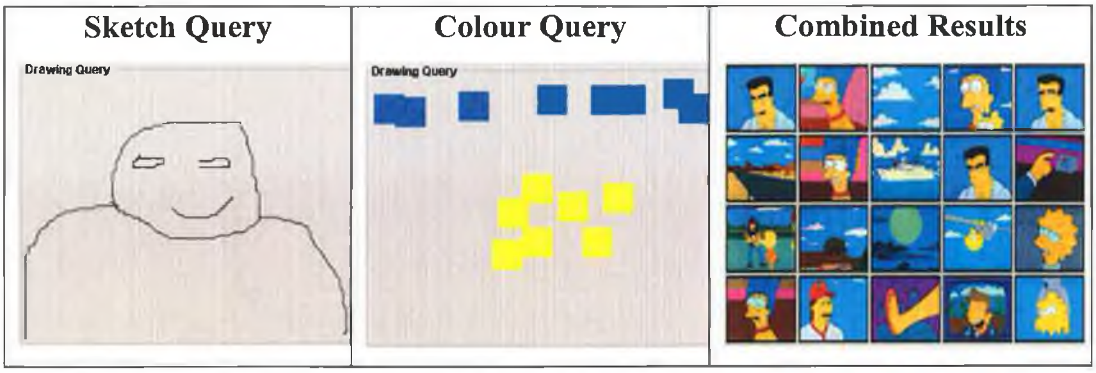


Figure 3-3: Example of Sketch-based Query

Figure 3-3 shows how a user's drawing skill can be somewhat limited and it remains to be seen if a sketch-based interface can perform well for most users. The displayed results do contain similar colours in the same areas as the query while their shapes are also 'somewhat' like the drawing.

This method of search is again very much query dependent. If an individual is searching for "Video of Bill Clinton" then drawing a sketch will be of little help and a colour search won't provide any real benefit either as colour isn't a real factor in the query. However if the search is for "Boats at sea" or "Soccer Field" then because valid results would be expected to contain strong backgrounds of "Blue" and "Green" respectively we would expect high retrieval performance.

[5] Concept-based search: The user can search video-based on specific concepts. As discussed in the previous chapter there are numerous concepts that that could be available from 'aeroplane', 'car' and 'building' visual concept feature detectors to 'speech', 'music' and 'monologue' audio feature detectors. TRECVID features evaluation of systems with various concept-based search facilities.

A specific concept will generally come with a confidence weighting value that can be used in the ranking process and the user simply decides to include a concept if it matches their information need. As concepts tend not to be very specific their benefit is dependent on the type of query/ information need of the user. Provided the concepts match the information need there would be an expected benefit for retrieval. For example, if the user's query search is for "Ducks on a pond" and the three concepts they can use are "Faces", "Planes" and "Music" then the concepts are likely to be of little use for that query. However, if the following two concepts were also available "Water" and "Outdoors" then we would expect a performance benefit.

3.1.2 Object-based Search in Video IR

Another possible method of video IR is object-based retrieval. Currently there is no 'complete' object-based retrieval available that uses full object information in the retrieval process. The closest available is concept-based search which offers retrieval on a limited number of objects like cars, buses and planes but with limited user interaction apart from selecting or de-selecting the concepts for specific queries. Object-based concepts generally fail to include additional attributes like location and size that might prove useful in the ranking process.

In object-based search the user can specify an object within an image for inclusion in the ranking process instead of providing a complete image as an example of their information need. Additional object-based attributes could also prove useful for the relevance feedback model of retrieval that we discuss next.

3.2 Relevance Feedback in Text Retrieval

In this section we will discuss the relevance feedback (RF) model for **text** IR. The concept of relevance feedback is actually quite straightforward. When users view ranked list(s) of results they include/ exclude the ranked documents in an expanded query which is used to generate a new ranked list and the cycle continues.

The user initiates a query search-based on some specific or vague information need by inputting a number of terms into the IR system. The document index is searched and a ranked list of documents is returned to the user. Up to this point the approach is the same as most standard text IR search engines. Where the RF approach differs is that the user can judge documents from a returned ranked result as being relevant or non-relevant to their information need and these can be included in their query, having made document judgements, the results can be re-ranked accordingly. As documents are judged by the user additional document terms are included and removed from their query automatically.

The RF process is actually an iterative one. For the first RF iteration the user ranks documents by providing search terms. Each subsequent iteration occurs after the user has judged N documents as being relevant or non-relevant to their information need. At each RF iteration the user's query is refined and should get closer to their actual information need.

3.2.1 Rocchio's Algorithm

The first relevance feedback for text IR was devised by Rocchio in the late 1960's with a popular reprint of his work republished in 1971 [Rocchio, 1971] [Harman, 1992]. He developed a relevance feedback approach with a ranking algorithm that incorporated a weighting scheme for positive and negative document judgements. Rocchio's algorithm operates as follows:

$$Q_1 = Q_0 + \beta \sum_{i=1}^{n_1} \frac{R_i}{n_1} - \gamma \sum_{i=1}^{n_2} \frac{S_i}{n_2}$$

Q_0 : This is the initial query term vector

R_i : The term vector for the relevant documents

S_i : The term vector for non-relevant documents

n_1: Number of relevant documents

n_2: Number Non-Rel documents

Relevant and non-relevant judgements are not generally weighted equally, and so β and γ are used to adjust the weighting of relevant and non-relevant judgements respectively. While there are no commonly agreed values for β and γ, some research has indicated that a value of 0.75 for β and 0.25 for γ seem to perform best [Salton and Buckley, 1990] [Harman, 1992]. A different weighting approach is used for relevant and non-relevant judgements because documents contain both positive and negative aspects, assuming that no document is a perfect example of either.

We know a ranked document contains strong positive aspects because the user has indicated it as such via keyword terms and judgements. When the user makes a negative judgement on a document that is ranked highly it must contain a reasonably high positive aspect in terms of the relevancy of users' given keywords, assuming the text ranking process is efficient and the user has provided correct keywords.

Rating a ranked document as non-relevant represents a challenge as the whole document is generally marked as non-relevant by the user even though a number of the terms would undoubtedly be relevant, and perhaps only a small number of terms are actually non-relevant. As a result of this we can expect that positive judgements will be given a higher weighing confidence than non-relevant judgements and this is reflected in Rocchio's algorithm.

3.2.2 Relevance Feedback Enhancements

Research into RF has been ongoing for a number of years with demonstrated improvements over non-RF techniques on various evaluation corpuses from CRANFIELD [Cleverdon el al., 1966] to TREC (1993, 1994 and 1995) and was studied in detail by Donna Harman [Harman, 1992].

There have been a number of improvements made to Rocchio's original RF algorithm apart from modifications to the original weighting scheme. Originally RF was developed for text IR's vector space model, a variation introduced by ([Robertson and Sparck Jones, 1976]) for the probabilistic IR model in 1976. Another interesting technique developed by ([Robertson el al., 1992]) took selective terms in documents for the re-ranking process instead of the document as a whole and showed improvement over the standard approach of using all document terms. The term weighting is based on the number of term occurrences in relevant and non-relevant documents. A 'good' term is one that has a higher number of occurrences in relevant documents as apposed to non-relevant documents. Another approach by [Buckley and Salton, 1995] used TREC development data to train their system to weight documents in a dynamic fashion instead of the traditional approach of static weights for documents retrieval. The weight of each term was adjusted and the results evaluated on the TREC learning corpus until an optimised weight was obtained for all terms, terms from future RF experiments use the optimised weight.

Pseudo Relevance Feedback

A technique called pseudo relevance feedback has become popular in recent years. Pseudo RF uses an artificial ranking mechanism instead of real user judgements in the weighting process. In lieu of the user explicitly making judgements on the re-ranked results the system automatically includes the top N documents as relevant, re-ranks the results and after N iterations the results are returned to the user [Xu and Croft, 1996].

The benefit of this technique is that the user does not have to continually refine their query, saving time and effort. The pseudo RF algorithm can also be evaluated automatically without the need for slow and complex user testing. The disadvantage is that the technique is only an approximation of user behaviour and therefore performance can suffer slightly compared to true RF systems, but despite this, pseudo RF does provide a performance boost.

Ostensive Relevance Feedback

Ostensive RF is an interesting relevance feedback technique for text that takes into account the relative age of document judgements received by the system where "age" is measured in terms of the user's current search session and an older document judgement is weighted lower than more recent judgements.

The Ostensive RF technique factors the impact of a user's search over time, as documents are returned and viewed by the user they become influenced by their content and modify their query as a result. The Ostensive RF process, like many user's web search queries, is an iterative one with query terms added and removed at each iteration until valid document(s) are found that match the information need.

The main question with Ostensive RF is how best to weight older judgements. Usually some form of logarithmic decay curve is used to weight older user judgements. Figure 3-4 illustrates two decay weight possibilities:

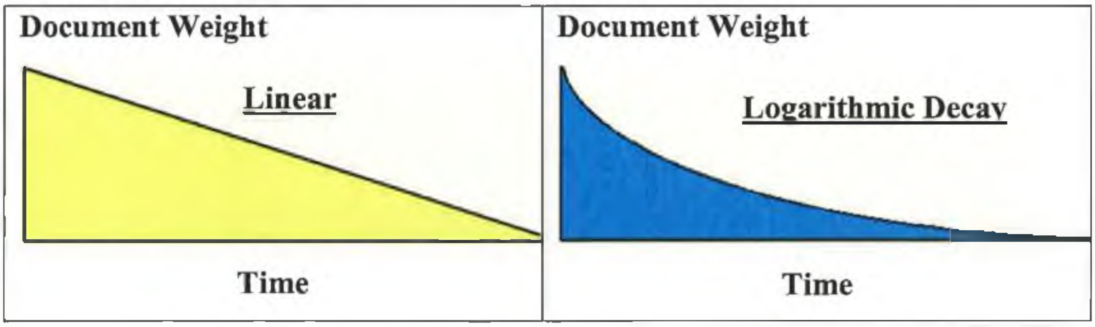


Figure 3-4: Ostensive Decay Curves

In Figure 3-4 we can see that the Linear Decay of document judgements is gradually constant, and the effect is a slow change in document weight over time consistent with a user's piecemeal query formulation. This means that more documents can be effective in the ranking process for longer. Table 3-1 below shows how text weights are decayed linearly from 1.0 to .19 percent after 10 iterations.

Table 3-1: Linear Ostensive Decay Rates

RJ Time	RJ Weight	RJ Time	RJ Weight
1	1.0	6	.55
2	.91	7	.46
3	.82	8	.37
4	.73	9	.28
5	.64	10	.19

The second example 'Logarithmic Decay' illustrates a larger document decay weight for newer document judgements; the decay rate reduces quickly initially with the reduction becoming more gradual over time. This means that the latest documents are the most effective in the ranking process with older documents becoming less important quickly. Table 3-2 below shows how text weights are decayed using logarithmic decay from 1.0 to .10 percent after 10 iterations.

Table 3-2: Logarithmic Ostensive Decay Rates

RJ Time	RJ Weight	RJ Time	RJ Weight
1	1.0	6	0.309
2	0.89	7	0.24
3	0.62	8	0.18
4	0.472	9	0.136
5	0.385	10	0.10

[Campbell, 2000] demonstrated an ostensive relevance feedback technique on a collection of annotated images. His model offered two methods of retrieval, traditional search and a specialised connected web graph interface that shows the user a visual representation of the retrieved documents. As the user moves the mouse over areas of the graph the nodes of connected images expand and disappear. The movement of the mouse over the connected nodes is taken as a passive relevance judgement.

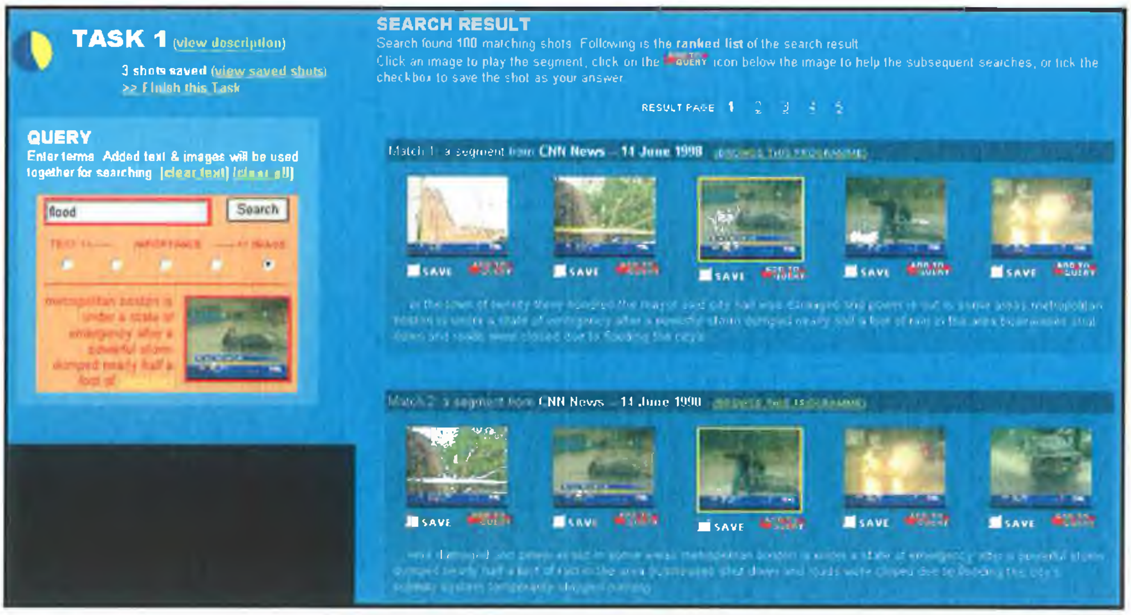
3.3 Relevance Feedback for Video Retrieval

The Relevance Feedback approach has been successfully applied on video retrieval [Wang et al., 2001]. As discussed in earlier chapters, video retrieval performance is very low in comparison to text with the level of semantic extraction possible from the visual layer being the main cause. RF for video could help overcome poor retrieval performance by users providing a number of valid and invalid examples to the system, and with this feedback the system could over time provide an improved ranking [Browne et al., 2003].

Pseudo relevance feedback has also been applied to video IR in the Informedia project (see Section 1.6.2). The approach takes an initial query input from the user, usually in the form of an image or a number of text keywords, much the same way as a traditional video IR system. The system ranks content accordingly and automatically takes the top *N1* video results as query inputs, and after *N2* iterations the results are shown to the user. This type of approach is evaluated each year in TREC with groups using the supplied text and images to start a pseudo RF search.

3.3.1 Current RF Video Retrieval Systems

A recent example of a Relevance feedback video IR system was the TREC 2003 system developed by Dublin City University [Browne et al., 2003]. The system incorporated positive text and video-based relevance feedback with video shot ranking. The facility was offered for the user to modify their RF judgements to favour text, video or both in the ranking process. The user's initial query search could include visual query topic examples and text supplied by TREC and when the results were ranked/ re-ranked they could also be added as feedback.



TASK 1 (view description)

3 shots saved (view saved shots)
>> Finish this Task

QUERY
Enter terms. Added text & images will be used together for searching [clear text] [clear all]

flood

SEARCH RESULT
Search found 100 matching shots. Following is the ranked list of the search result.
Click an image to play the segment, click on the ADD TO QUERY icon below the image to help the subsequent searches, or tick the checkbox to save the shot as your answer.

RESULT PAGE 1 2 3 4 5

Match 1: a segment from CNN News – 14 June 1998

Match 2: a segment from CNN News – 14 June 1998

Figure 3-5: Dublin City University TRECVID 2003 Screenshot

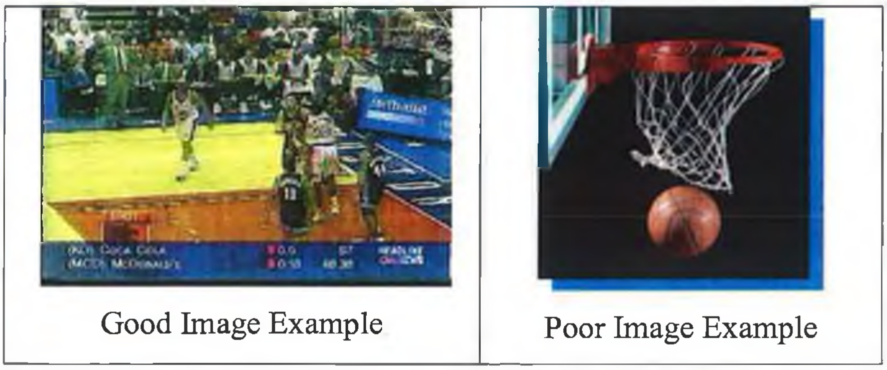
Figure 3-5 shows a screenshot of the relevance feedback video IR system. The shots with user relevance judgements are displayed on the left and are included in the subsequent query process. The newly ranked results are displayed on the right, and each result provides an "add to query" button. By clicking on this a relevance judgement is made by the user and the corresponding shot is added onto the image query list on the right. Weighting of text keywords and images can be adjusted via the checkboxes shown on the left.

Another interesting feature with this system is that the previous and next two linear shots are shown before and after each ranked result. This is important as in many cases relevant shots can be found occurring in the video around ranked results.

3.3.2 RF Video Retrieval Performance

It is perhaps even more important to include Relevance Feedback for video retrieval than for text retrieval. RF for video offers the possibility of partially overcoming video's major IR barrier of poor feature discrimination as more visual examples can be included. Document text generally contains structured semantic information in large quantities, but with video however it is a different story as only a small amount of semantic information is available especially from the visual layer.

As discussed in the first chapter, the quality of visual query examples that are used for video retrieval can have a big impact on retrieval performance so it stands to reason that including more quality examples via an RF process can improve performance still further. Figure 3-6 shows two topic example images given for the TRECVID 2003 query "*Basketball Matches*". The first image example is considered 'good' for retrieval while the second is considered 'poor'.



Good Image Example

Poor Image Example

Figure 3-6: Good and Bad Visual examples

There are two main reasons why the 'good' image example shown above should perform well when used as a query image, the first is that it is taken from the same type of video source that is available for retrieval namely TV news. This means there is more chance of finding close matches in the content. The second is that the background in terms of both colour and edges is very similar to the source so valid results should also rank quite high.

The poor example above is such because it fails on both those points. The image contains a large black background which is quite dissimilar to source content. The remaining colour and edges contained in the image are also unlikely to find close valid matches in the content.

A relevance feedback and QBE system's performance benefit is critically dependent on the quality of the image judgements given "poor judgements in -poor results out". It can be a challenge selecting a representative query image for QBE or RF.

If the user chooses a good query image example it can help point the video IR system in the right direction, and this is especially important for low-level features like the colour histogram which are dependent on representative image examples to perform well.

RF requires increased interaction and decision making from the user in order to be effective and there is a limit in the amount of this time and effort we can expect from the user. Choosing 'good' positive and negative judgements requires an element of training and experience and this will be shown in Chapter 5.

Some research has been done on the performance benefit of visual RF over a number of iterations and has shown a limit in real performance after 6 iterations on image-based retrieval [Heesch and Rüger, 2003].

3.4 Summary

In this chapter we started looking at traditional video IR in terms of search possibilities discussing text-based, query by visual examples, concept-based and user sketch queries. Video IR can use combinations of these during retrieval as each offers their own advantages and disadvantages. Video is an audio-visual medium and requires combination of approaches in order maximise performance.

One of the most popular methods of video IR is text-based search. The video text is extracted from two main sources, associated closed captions (subtitles) or the spoken audio. A disadvantage of a text-based search on this is that users will not always mention their location, what objects are visible, their names or what event is occurring at the time. Therefore a text only search can miss relevant results.

The chapter discussed the Relevance Feedback technique for text retrieval. The idea behind this approach is that as documents are ranked, the top N results can be judged relevant or non-relevant by the user and incorporated into an expanded query. The process is iterative and as it continues the query is gradually refined getting closer to the user's information need. An automatic approach called pseudo relevance feedback can also be used which does not take user input but automatically includes the top N documents at each iteration. The technique and its variations have been shown to be successful by a number of researchers.

RF has also been successfully applied to video and image retrieval in both automatic and user-based variations. It helps to overcome the limitations of current visual features by including numerous relevant and non-relevant examples, and, as discussed in Chapter 1, good examples can have a big impact on retrieval performance.

An interesting text-based RF variation known as ostensive relevance feedback was also discussed. This approach takes into account the age of document judgements in terms of the user's search session in the weighting and ranking process. The idea is that as a user starts a query search their information need is vague and somewhat undefined, but as time progresses they are influenced by the ranked results and their query becomes more refined. Ostensive RF factors this into the retrieval process.
This RF technique will be discussed in more detail in the next chapter where an ostensive RF model for video will be presented which includes object information.

Chapter 4 An Ostensive-based RF Model for Video IR incorporating Object-based retrieval

4.1 Introduction

In the previous chapter we discussed IR and various relevance feedback (RF) approaches which incorporated users/ automated judgements into the retrieval process. We noted the importance of RF for video given the limitations of current video retrieval performance. The previous chapter also detailed a variation of RF for text retrieval known as ostensive relevance feedback. This approach had previously been evaluated on the retrieval of text captions from annotated historical French images with some positive indications.

In this chapter we will discuss an ostensive RF model adapted for video based IR that incorporates object-based feature information. With this 'ostensive' approach, the relative[3] time of the user's judgements in terms of their overall retrieval session, or rather the rank ordering of relevance judgements, is used as a factor in the ranking process. The principle is that a user's older or earlier shot relevance judgements (***RJ***) are given reduced weighting in comparison to newer and more recent judgements. The concept behind this approach is that, as people view content and make selective relevance judgements, knowledge of their own information need increases and therefore newer and more recent judgements are given a higher weighting than older ones [Bates, 1989].

An ostensive RF model for video has to deal with numerous and complicated issues including overall video structure as shots are not independent units unlike documents, various feature type combinations, value comparisons and shot ranking. This chapter will discuss solutions to these problems.

[3] Relative time is concerned with the rank ordering of a user's relevance judgements with later judgements given a higher weighting.

4.2 Unit of Retrieval

The first task we will examine is how to decide on the unit of retrieval. Chapter 1 discussed the main structural components of video in terms of frames, shots and scenes. Most frames from individual shots are too similar to each other to be considered separately for effective retrieval. On the other hand, scene based detection/ retrieval cannot be performed reliably on most genre types (television news is the main exception). Therefore, scenes too cannot be considered.

This leaves video shots as the most appropriate unit for retrieval, and from previous chapters we have seen that shot-based retrieval is the most popular retrieval unit for video IR-based systems. The reason for this is that shots can be detected with very high accuracy in comparison to scenes and yet are small enough to be representative of information content without containing too much visual redundancy. In this model video shots are the unit of retrieval with various indexed features aligned at the shot level.

4.3 Overall Ranking Considerations

An Ostensive RF Model for video content will need to incorporate various important attributes in order to provide effective video IR ranking. The following are what we believe to be the five main requirements:

[1] User-based relevance judgements on video. The model will need to be able to accept positive and negative RF judgements from users on **n** video shots where **n** can vary from 1 to ∞. A user can supply any number of positive and negative shot examples that all require to be included in the ranking process.

[2] Low level visual features. The model will need to combine a variety of different low-level features in the RF retrieval process from users shot judgements. The output from visual feature analysis can vary dramatically with completely different

properties and ranges of values and so cannot be simply combined. Imagine we decided to combine 9 colour features and 1 edge feature clearly a straightforward combination will be colour biased.

[3] Text-based closed captions. For most shot there are usually some associated text subtitles (closed captions) which need to be considered for RF query and ranking.

[4] Object-based Information. User relevance judgements can include specific objects from a shot as positive or negative RF and these also need to be included for ranking.

[5] Ostensive weighting. The final aspect that needs to be considered is the relative temporal weighting of users' RF shot judgements. Older judgements should be given a reduced weighing compared to later selections. There are a variety of different scales with which relative weighting can be calculated as was seen earlier in Section 3.2.2.

4.3.1 Shot Query States

During a user's query session, at any point in time the user query (**Q**) in the case of our system can be a series of *relevance judgements* rather than just a fixed statement of the user's information need, and can have textual (**Tx**) and/or visual (**V**) and/or Object (**Oe**) components. The initial user query may contain required keywords from the content of the ASR transcript, low-level features or specific objects. The following are the possible query states (ranking) for Visual, Textual and Object-based features that could occur:

[1] Text-based query (Tx): The user has entered **Text (T)** as their query to start or expand the query.

[2] Visual-based query (V): The user has selected shot-based visual features to start or expand the query.

[3] Object-based query (Oe): The user has selected an object from a shot to start or expand the query.

[4] A combination of [1], [2] and [3]: The user selects text and/or Visual and/or objects to start or expand the query.

As each of these states is reached (after each relevance feedback iteration) the overall results are ranked or re-ranked and presented to the user. Table 4-1 gives an indication of the various user query combinations, with '1' indicating the option is used whereas '0' indicates it is not.

Table 4-1: Query Ranking scenario truth table

Scenario Num	Tx	V	Oe
1	1	0	0
2	0	1	0
3	0	0	1
4	1	1	0
5	1	0	1
6	0	1	1
7	1	1	1

In order to increase retrieval performance non-relevant text is not considered.

4.3.2 Ostensive Ranking Considerations

The ostensive weight of users judgements is based on the order in which they are chosen and can consist of any of the following:

- Text Search
- Relevant shot (four low-level visual features discussed in Section 5.1.4)
- Non-relevant shot (four low-level visual features discussed in Section 5.1.4)
- Relevant object
- Non-relevant object

As the user makes each judgement it is automatically assigned an ostensive weight of 1.0 and the weights of all previous relevance judgements are reduced. The ostensive weight for relevance judgements is based on the order (time) in which they are chosen.

Ostensive Weight Example:

- The user has chosen 2 relevant shots **A** and **B,** and 1 relevant object **C**
- **A** was selected first and assigned a weight of 1.0
- **B** was selected second and assigned a weight of 1.0, now **A** has a weight of 0.9
- **C** was selected third and assigned a weight of 1.0, now **A** has a weight of 0.8 and **B** has a weight of 0.9.

4.3.3 Shot-to-Shot Similarity Ranking

In this section a general overview is given of shot to shot similarity computation for ostensive RF-based retrieval. The main requirement for RF is that ***all*** shots with positive and negative relevance judgements to be compared against ***each*** shot in the corpus before generating overall ranking results and returning them to the user.

As we discussed previously, a user can select low-level features and/or text and/or object information from shots as part of their positive and negative judgements, and these will need to be combined and ranked before being returned to the user in two steps. Each of these feature types is first considered separately in the shot comparison process and will be discussed in more detail later in the chapter. A weighting factor is then given to each feature and the final ranking is therefore obtained by the linear sum of the features.

An important aspect that needs to be included for all types of feature ranking in shot retrieval is the comparison approach, namely whether it is a similarity or dissimilarity comparison that we are dealing with. For *similarity* comparisons the higher the value the closer the match therefore results are ranked in descending order before being displayed to the user. For *dissimilarity* comparisons the smaller the value the closer the match therefore results are presented to the user in ascending order.

Firstly, we show a visual shot comparison method using the dissimilarity between shots. Assume a normalised feature vector F of a given shot S has k values and F_i (S) indicates the *i-th* feature value in the vector of shot S. Given any two shots S_a and S_b, the dissimilarity can be obtained based on the sum of absolute difference (**abs**) between each pair of feature values $F_i(S_a)$ and F_i (S_b) $(1 \leq i \leq k)$, as shown in Formula 4-1 below. For a relevance feedback process, one of the shots will be a user's judged shot and the other one will be a shot from within the corpus. Absolute distance can be considered equivalent to Manhattan difference.

$$\text{Shot\_Dissim}(S_a, S_b) = \frac{\sum_{i=1}^{k} abs(F_i(S_a) - F_i(S_b))}{F_i = i^{th}\ Feature} \quad \textbf{(4-1)}$$

Table 4-2: Absolute Difference Example

Bin Num	Histogram1	Histogram2	Absolute Score
1	200	125	75
2	100	170	70
3	50	90	40
4	25	15	10
Total			195 non-similar

Table 4-2 shows four histogram bins (1-4) being compared using an absolute difference method. The four bins each represent different and non-overlapping colour ranges while the histogram values represent the number of image pixels in the particular range. The final total of 195 is a simple total of all absolute scores. For the purpose of presenting the shot-to-shot similarity formula we assume that features such as colour histograms are derived from the entire shot but later we will show an approximation to this by using features only from keyframes.

The second method for shot-to-shot comparison is to calculate the similarity between two shots and the final score is based on the sum of the *intersection* of feature values F_i $(1 \leq i \leq k)$ from two given shots S_a and S_b. Instead of computing the difference between two feature values the number in common is computed.

$$\text{Shot\_Sim}(S_a,S_b)=\sum_{i=1}^{k} Min(F_i(S_a),F_i(S_b)) \qquad \textbf{(4-2)}$$

$$F_i = i^{th}\ Feature$$

Table 4-3: Intersection Difference Example

Bin Num	Histogram1	Histogram2	Intersection Score
1	200	125	125
2	100	170	100
3	50	90	50
4	25	15	15
Total			290 Similar

Table 4-3 shows an Intersection difference example, where differences between values scores are calculated based on the common values. As with the previous example the four bins represent colour ranges while the histogram values represent the number of image pixels in the particular range. The final total of 290 is a simple total of all intersection scores.

Unless otherwise stated all low-level visual feature comparisons (i.e. Colour Histogram, Colour Region average, Edge Distribution and Edge Shape) will use the absolute difference method. Other features like text and object templates require a different approach and they will be further discussed in Sections 4.4, 4.5 and 4.6.

4.4 Text-based Ranking

In retrieval of video shots, text is often made available and this can be derived from the spoken dialogue transcript, from text appearing on-screen as part of the image, or from a description of what is happening in the video. In our work we assume that text is derived from the spoken dialogue via either an automatic speech recognition (ASR) or from closed captions (CC) and in this section we will discuss how text associated with a video shot can be incorporated into our model of retrieval.

Before extracted text can be used for retrieval it will need to have standard information retrieval techniques applied to it like stemming and stop-word removal.

A comparison between two shots based on associated text is done by matching common terms allowing the calculation of a ranked similarity score for each shot. In text retrieval a popular term weighting method uses a combination of term frequency **(tf)** times the inverse document frequency **(idf)** see Chapter 3 Section 3.1.1.

As we discussed in Chapter 3, Section 3.1.1, there are different document ranking algorithms that could be employed in video shot retrieval but for the purposes of text retrieval with this ostensive RF model we will use the TF-IDF approach to weight and rank video text. The similarity $Shot\_Sim^{text}$ between any given two shots S_a and S_b can be calculated based on the dot product between their term vectors W_a and W_b. where j is the order position of the j-th index term in the vector. Equation 4-4 shows shot similarity for text with ranking based on the result.

$$Shot\_sim^{text}(S_a, S_b) = \sum_j W_{aj} * W_{bj} \qquad \textbf{(4-4)}$$

In our work we further extend this document ranking approach to shot retrieval by including a factor to account for ostensive decay weight of the relevant judgement shots.

The TF-IDF formula describes how the indexed term weightings are calculated and this is used to generate a shot comparison score. The shot comparison score and ostensive decay weight is computed at run time based on user relevance judgements and the order of the judgements. Formula 4-5 below ostensively weights shot comparison score *shot_sim()* of a RF shot S_i^{rel} and a candidate shot C_j in the corpus based on the ostensive decay weight OD_i of S_i^{rel}. $Shot^{decay}()$ shows the comparison score between S_i^{rel} and C_j when the i-th relevance judgements is made.

$$shot^{decay}(S_i^{rel}, C_j) = shot\_sim^{text}(S_i^{rel}, C_j) * OD_i \qquad \textbf{(4-5)}$$

OD is the ostensive decay of the shot similarity value. The weight ranges from 1 to 0 with the weight reduction based on the order of the user judgements. A number of possible decay rates are discussed in Chapter 3 Section 3.22.

The calculation of shot comparison values using text information is reasonably straightforward when we are comparing two shots, even when accounting for ostensive decay, but how does one handle a query which compares text from a number of RF shots against each remaining un-judged candidate shot in the corpus, as we might expect in a real usage scenario? For example, if a user has already viewed a number of shots and found *k* to be relevant and *l* to be non-relevant, then according to our approach we need to compute the similarity between each of these relevant and non-relevant shots, and the original query, against each of the as yet unseen shots in the corpus.

Because each unseen shot in the corpus (S_1 to S_n) needs to be compared against *all shots with relevance judgements*. This will require the comparison of **n^x (k+l)** shots against each $\boldsymbol{S_n}$ shot before a total score for text would be available ($\text{Total}^{\text{Text}}$), given k relevant judgements and the original query, for non-relevant judgements the same applies. To calculate the overall similarity for the ***Sn*** shot against the ostensive judgement, a score is obtained for each relevant (X_{rel}) and non-relevant ($X_{non\text{-}rel}$) shot with the result placed in one of two totals: (1) the relevant total or (2) the non-relevant total. When all judged (**RJ**) shots have been compared against a candidate shot, the $Total^{Text}$ score is calculated by subtracting the non-relevant total from the relevant total.

$$Total^{Text}(C_j) = \sum_{i=1}^{k} shot\_sim(X_i^{rel}, C_j) * OD_i - \sum_{m=1}^{l} shot\_sim(X_i^{non-rel}, C_j) * OD_m \quad \textbf{(4-6)}$$

$X_i^{rel} \in \mathrm{X_{rel}}$

$X_i^{non-rel} \in \mathrm{X_{non\text{-}rel}}$

Cj :	a candidate shot in the corpus
X^{rel} :	a shot with Relevant Judgement
OD_i :	Ostensive decay weight of RF shot.
k :	the total number of shots with Relevant Judgement
$X^{non\text{-}rel}$:	a shot with Non-Relevant Judgement
l :	the total number of shots with Non-Relevant Judgement

OD is the ostensive decay weight of the shot similarity value. The weight ranges from 1 to 0 with the weight reduction based on the order of the user judgements. A number of possible decay rates are discussed in Chapter 3.22.

When all shots in the corpus are compared against the RJ shot, a final shot ranking can be produced and returned to the user. However should the query contain visual and/or object-based-elements then totals must be produced for these aspects also before generating a final ranking.

In this section the computation of $Total^{Text}$ has been discussed, and the remaining sections of this chapter will outline the calculation of $Total^{Visual}$, $Total^{Object}$ and their relative weights W_1 to W_3 as used in the overall shot ranking formula. The following is that overall ranking formula which we will use:

$$Rank(C_j) = W_1 * Total^{Text}(C_j) + W_2 * Total^{Visual}(C_j) + W_3 * Total^{Object}(C_j) \tag{4-7}$$

4.5 Visual Comparison Ranking

The previous section of this chapter discussed how the model for video shot retrieval developed in this thesis provides text-based ranking on video content. In this section we will detail how visual features can be incorporated into the ranking and retrieval process. The visual features that will be used in the model are the following four low-level features: Colour Histogram, Colour Region average, Edge Distribution and Edge Shape analysis. These algorithms were discussed in Chapter 2 in Sections 2.2.3 and 2.2.5.

Images require different approaches to indexing, comparison and ranking than text. Indexing a video 'shot' could require ***n*** individual still images to be analysed for a shot of n frames at 25 frames per second. Each of these f_i $(1 \leq i \leq n)$ images could need to be analysed by anything up to k visual features, as no single visual feature is sufficiently powerful enough to provide broad and accurate visual retrieval. These requirements generate too large an index to provide efficient retrieval and the problem is generally simplified, as we will see in the later sections.

4.5.1 Visual Shot Representation

A single video shot is constructed of ***n*** images with a fixed number of frames per second of video (typically 25 for Europe, Asia and South America and 30 for USA and Japan). The first problem we address here is how to index all of these images. There are many ways in which we can address this including the following five approaches:

[1] Take X_n still images from the RF shot (RF^1) and compare these against each frame Y_k in the candidate shot (CAD^1). The comparison values $(n * k)$ can be averaged to produce the resulting overall shot score. Note the formula is symmetric, *shot_dissim*(RF^1, CAD^1) = *shot_dissim*(CAD^1, RF^1).

$$shot\_dissim(A,B) = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n} Compare(A_i, B_j)}{k * n} \quad \textbf{(4-8)}$$

This method of representing a shot is computationally expensive and just because **all** frames in a shot are included in the similarity does not mean that they are all equally important. For example, in a shot of 20 seconds of a landscape with nothing happening followed by 3 seconds of a person walking into the frame, the first 20 seconds will dominate the representation of the shot whereas clearly the most important part is the final second.

[2] In a second approach which tries to assign some degree of importance, we use each image in the comparison process as before but remove the highest and lowest 10 percent of the values before finding the average. If there are 50 frames in shot A, and 60 frames in Shot B, the total number of individual comparison values is 3,000 (50*60), the highest and lowest 10 percent of the

values are 300 each, respectively. Thus the new total: 3000 – 600 = 2400. This removes possible outliers from contributing to the overall shot score but this may actually be a bad thing as the example in [1] shows.

[3] Thirdly, we could just use the average of the top 10 percent of individual image comparison scores as the shot comparison result and what this does is that it allows one shot focus to be matched against the other shot focus, which is good.

[4] A fourth approach to shot-shot comparison is to remove the top and bottom 10 percent of image comparisons, average the remaining scores and find the pair of images in both the RF shot and the comparison shot that is closest to the average. This is equivalent to finding the closest frame in shot A to shot B and vice versa.

[5] Finally, a popular approach is to index only a representative 'keyframe' image for each shot and use this keyframe as the sole representative of the shot. This frame can be computed as the first, the last, the one most similar to all the others in the shot, but generally it is usually taken as the frame from the middle of a shot so as to avoid frames from neighbouring transitions and serves as visual representation of the shot.

With the first three options for shot representation a single 'virtual' image is essentially created. The 'virtual' image feature results can be added to the index, thus reducing storage requirements. The problem with these approaches is that an 'averaging' of features can in no way guarantee that a more representative 'virtual' image is produced. The most straightforward approach is to use the fifth option, the keyframe, and to index a single keyframe only as the 'representation' of the shot. This approach can be used for visual features like colour and edge histograms and regional averages but it is not sufficient for temporal-based approaches like camera motion detection.

4.5.2 Normalising Visual Features

In shot-to-shot comparison, depending on the query, there might be a combination of visual features to be included in the calculation, each of which may have their own unique set of attribute values. These feature values need to be combined before an overall $Total^{Visual}$ score can be computed for the shot-to-shot similarity and this is achieved by normalising the results.

Naturally there are plenty of ways in which this can be achieved. One basic approach that can be employed is an averaging of feature results whereby the shot comparison value for each feature is added to a total and divided by the number of features. This approach ignores any large differences between the features as Table 4-4 shows:

Table 4-4: Shot A comparison with RJ and shot B

Feature	**Comparison Score**
Colour Histogram	1000
Colour Region Average	50
Colour Region Median	50
Edge Histogram	100
Comparison Total	**1200**
Comparison Average	**300**

As we can see from the example above the colour histogram comparison value is so large in comparison to the other features that it effectively makes the others meaningless. The average comparison value (without colour histogram) 200 / 3 = 66.67 compared to 300 when it is included. If these values represent normal shot comparisons using these features then Colour Histogram will be completely dominant.

As the previous example has shown, an average might not be an accurate representation of the feature activity within shots, as a high or low value produced by an individual feature can skew the overall average. In an attempt to alleviate this problem a second method known as ratio averaging can be employed. The approach works by taking an average of shot comparisons for each of the features (1 to n). When these averages are obtained they are divided until they are 'approximately' similar to a base value of 100. For all future visual comparisons the scores are divided by the same values as was found for the averages. This approach does not guarantee exactness but it can provide a useful method of combining individual feature scores. Table 4-5 is an example of this second approach:

Table 4-5: Ratio Normalisation

Feature	**Total Score**	**Average**	**To Normalise**	Result
Colour Histogram	460000	920	Divide by 8.3	**111**
Colour Region Avg	55000	110	Leave	**110**
Colour Region Median	62500	125	Leave	**125**
Edge Histogram	28000	56	Divide by 0.5	**112**

Now that we have the ratio calculation we can combine the four features. Table 4-6 is an example of a visual shot-to-shot comparison using absolute difference. Absolute Difference is calculated by taking the smallest value from the largest.

Imagine that a colour region value from one shot was 130 and from the other it was 180, the absolute difference would be 50. If the first shots value was 200 and the second was 150 the answer would still be 150.

Table 4-6: Ratio Example

Feature	**Comparison Value**	
Colour Histogram	870	870 / 8.3 = 105
Colour Region Avg	65	
Colour Region Median	140	
Edge Histogram	72	72 / 0.5 = 144

The normalised *score* for this shot-to-shot comparison is 105 + 65 + 140 + 72 = 382

Our model in this thesis will use the Ratio Averaging approach to normalise and combine the low-level visual features.

4.5.3 Visual Shot Comparison

At this point we have discussed how video shots can be indexed efficiently based on visual aspects and various visual feature values normalised and combined. In this section we will discuss how shot-to-shot visual comparison values can be computed.

As we described in the previous section, absolute distance will be used to compare differences between shot features. In practice, this requires each value of a feature vector to be compared and their differences totalled to produce a shot score. The score will be normalised as described in previous section before giving any ostensive weighting. For the next step an ostensive decay needs to be considered for this totalled score, remember that this is a ***shot dissimilarity*** so reducing the value will actually indicate the two shots are more similar. Instead of reducing the comparison value, ostensive weighting will actually increase it by multiplying it in the range of 1.0 to 2.0 after each iteration of the user's search.

A range of 1.0 to 2.0 was chosen to reduce the impact of an ostensive increase in comparison scores as the range examples discussed later illustrate.

An ostensive decay curve is used to indicate the ostensive weighting and how the weight values change with each iteration of the search. The curve is created based on either a linear or logarithmic function and the relative time of the shot judgement as made by users is the prime parameter in this function. Another parameter we use is the selection of the proper multiplication range. Deciding on the multiplication range is based on analysing the average comparison values. If the range was set from 1.0 to 10.0 then after another RJ shot was added by the user, the previous RJ's score would double, effectively negating its value in the ranking process.

If we use the score from the previous example, each of the four visual features produced the following normalised scores: 105 + 65 + 140 + 72 = 382.

> Range Example 1: If we consider that the RJ shot has an ostensive weighting of .80 and the range 1.0 to 2.0 then the score becomes: 382 * 1.20 = 458.4

Range Example 2: If we consider that the RJ shot has an ostensive weighting of .80 and the range 1.0 to 10.0 then the score becomes: 382 * 2.0 = 764

As we can see from example 2 after one RF iteration (one additional shot given a RJ) the visual comparison score effectively doubles. Clearly this is too much if we want more than one shot to contribute visually in a RF process. The first example is clearly less drastic and allows more shots to be effective in the RF process. How many effective shots is dependent on the ostensive decay rate that we choose. "Which rate is more effective?" is one of the questions we will discuss in the next chapter.

The following visual comparison equation is similar to the text equation described in Section 4.2.2 except that a different method of scoring is applied. The shots are ranked based on their overall visual score (features normalised using ratio averaging) before an overall ranking is obtained. The shots are ranked in ascending order and displayed to the user where further relevance judgements can be made.

$$Total^{Visual}(C_j) = \sum_{i=1}^{k} shot\_dissim(X_i^{rel}, C_j) * OD_i - \sum_{i=1}^{l} shot\_dissim(X_i^{non-rel}, C_j) * OD_i \tag{4-9}$$

A point to note however is that absolute difference will generate shot dissimilarity scores for the four visual features that we are including. Contrast this with the text scoring which generates similarity scores. This difference will create issues for combining shot rankings based on text and visual attributes. In order to solve this problem the text score is subtracted from the visual score in order to obtain a final overall ranking. The text score is similarity-based whereas the visual score dissimilarity-based therefore the text score needs to be subtracted from the visual score. The results are ranked in ascending order of dissimilarity and then displayed to the user.

Visual and Text ranking example when the i-th relevance judgement is made:

The visual comparison score for 4 visual features:

shot_dissim^{Visual} (x,y): (105,65,140,72) = 382.0
Ostensive weighting of **.80** (dissimilarity) range 1.0 to 2.0: 382 * 1.20 = 458.4
Text comparison score shot_sim^{Text} (x,y), 34.2 * .90 (Ostensive weight) = 30.78

Total Comparison Score: 458.4 – 30.78 = 427.62

Note: The visual shot relevance judgement was selected earlier than the text judgement and therefore the weight has ostensively decayed more than text. As the visual feature is a dissimilarity value, the ostensive decay of 0.2 is used to increase the visual score. The text score is a similarity score and therefore the ostensive decay reduces this.

4.6 Object Comparison

In this section we will discuss how object-based information is incorporated into the video shot retrieval model. While object features are visual in nature their attributes are sufficiently different from what we call "raw" features such as colour and texture, to necessitate a separate score calculation.

First let us discuss what the object information derived from video actually is. In the case of our work, what we refer to as object data is actually a template-based match which contains an object tag name, bounding box co-ordinates and a confidence matching value. Object-based template matching was discussed in Section 2.3 and involves searching video content for ***n*** known templates or in the case of our work, characters from the TV program, and if a 'close' match is found between one of the character templates and a shape occurring in the video then we record the location or position in the frame, size, and a value indicating how confident the matching process is. The example below shows the results from template matching on a frame of video from the animated television show "The Simpsons" which demonstrates the typically available attributes.

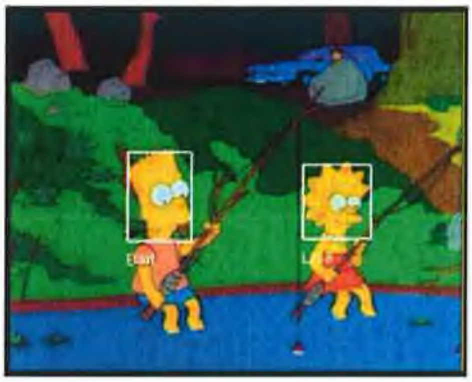


Figure 4-1: Object-based template matching example

The problem with the approach of creating and matching object templates is that there is an almost limitless number of objects that could be detected which makes template-based detection infeasible in the general case. However template matching can be effectively used to detect a small number of known objects which will be different from other objects occurring in the video. An example of this includes the major characters appearing in the Simpsons because they are all a similar colour (yellow) and different from other objects such as the tree trunks, car, boulders, etc. in the frame in Figure 4-1 [Browne and Smeaton, 2004]. In the next five sections we will discuss the main considerations for including object matching as part of a shot ranking model.

4.6.1 Weighting for a matching object

When considering how to assign a weight to the presence of an object in a matching shot, it is important to note that a given object will either be present or not (0/1); if it is present then a 'value' needs to be placed on a correct match between two shots with the given object, and this is chosen to be approximately equal to an average visual shot comparison. A matching object will be similarity-based and require an object value when a match is found but the difficulty is in deciding what the object value is. The average visual shot comparison value is 1400 and this will be the matching object value that we will use. This is somewhat ad hoc but for the present allows us to make progress with incorporating object matching into shot matching and seems intuitively sound. The following example shows how an object match can be computed:

Object Match Example:

Corpus shot ***x*** and relevance judgement shot ***y*** are being compared with the score based on visual and object-based features. The visual score for the shots is 2350 while a matching object score is **1400.** Ranking is in ascending order of dissimilarity. As an object match is similarity-based it will need to be subtracted from a visual score.

The visual score for shot_dissim$^{\text{visual}}$(x,y) = 2350
Between the two shots they have 1 matching object
The Total score: 2350 – 1400 = 950

4.6.2 Weighting for Multiple Matching Objects

The user may provide $\boldsymbol{n_p}$ positive and/or $\boldsymbol{n_n}$ negative judgements each of which has to be added or removed from a score in order to provide an overall shot similarity. This is similar to the previous example except that $\boldsymbol{n_p}$ relevant objects are subtracted from the score and $\boldsymbol{n_n}$ non-relevant objects are added.

Object Match Example 2:

Corpus shot ***x*** and RJ shot ***y*** are being compared. In this example the score is based on visual and object-based features. The visual score for the shots is 3000 while an object score is **1400** and ranking is in ascending order.

The visual score for shot_dissim$^{\text{visual}}$ (x,y) = 3000
A matching object is worth 1400
Between the two shots x and y, they have 2 matching objects.
y contains 1 non-matching object
The Total score: ((3000 – 1400) – 1400) + 1400 = 1600

4.6.3 Size of an Object Match

In ranking valid shots that contain matching objects, the similarity in terms of the size of the objects within the frames, could also be considered. For example, if five shots contain a matching object and the second and fourth shot are of a similar size or dimensions within the frame to the RJ comparison shot, then they could be ranked higher.

To accomplish this a 'size value' can be added to the object comparison value if two matching objects are also the same size. This 'size value' is taken as a percentage of the object comparison score. When comparing identified objects (A, B) from two shots (S1,S2) to see if they are in the 'same' size a threshold is used to allow their size to differ by +/- 10% and still be registered as the same size.

Threshold Example:

Object A and B have the following pixel co-ordinates in the image:

Object A	**Object B**
X1 = 50, Y1 = 50	X1 = 46, Y1 = 54
X2 = 150, Y2 = 120	X2 = 138, Y2 = 126

Width$_A$(50,150) = 100
Width$_B$(46,138) = 92
Height$_A$(50,120) = 70
Height$_B$(54,126) = 72
Difference(Width$_A$, Width$_B$) = 8 <= Threshold
Difference(Height$_A$, Height$_B$) = 2 <= Threshold

The width and height of two objects are less than a 10% difference threshold as a result a 'size value' can be applied.

Size scoring Example:

Corpus shot ***x*** and RJ shot ***y*** are being compared. Scoring is based on visual and object-based features with a visual score of 1650 and an object comparison score of **1400**. Ranking is in ascending order.

The visual score for shot_dissim$^{\text{visual}}$ (x,y) = 1650 *(dissimilarity-based)*
They have 1 relevant matching object
The matching objects have a similar size = 1400 * .1 = 140 *(similarity-based)*
The Total score: 1650 – (1400 + 140) = 110

This option is query-dependent and usage is based on the users information need, and it will not always need to be considered as for some user queries neither the size nor the location or position within a frame will be important.

4.6.4 Location of a matching object

Another factor in the shot ranking process which could be considered as part of shot matching is an object's location within the frames. If two matching objects are also in the same area of an frame then they could be ranked higher than matching objects found in different locations within a frame. For example if the RJ shot has the object located in the top right of the image then we may like to see valid results ranked higher if the matching object is also in the top right location. In order to facilitate this option a 'location value' can be added to the object comparison value if two matching objects are also in the same approximate area of the frame. This 'location value' is also a percentage of the object comparison score (usually 10 %).

When comparing identified objects (A, B) from two shots (S1,S2) to see if they are in the 'same' location in the image, once again a threshold is used to allow their locations to differ by +/- 10% and still be identified as the same location. This is similar to the previous size example except that only the first x,y co-ordinates need to be checked.

Location Threshold Example:

Object A and B have the following pixel co-ordinates in the image:

Object A	**Object B**
X1 = 50, Y1 = 50	X1 = 35, Y1 = 60
X2 = 150, Y2 = 120	X2 = 134, Y2 = 140

Difference(50,35) <= Threshold
Difference(50,60) <= Threshold

Since the co-ordinates are less than the 10% threshold a 'location value' can be applied.

Location Score Example:

Corpus shot **S1** and RJ shot **S2** are being compared. Scoring is based on visual and object-based features with a visual comparison score of 2000 and an object comparison score of **1400**. Ranking is in ascending order.

The visual score for shot_dissim$^{\text{visual}}$ **(S1,S2)** = 2000
They have **1** relevant matching object
The matching objects are in the same location = 1400 * .1 = 140
The matching objects have a similar size = 1400 * .1 = 140

The Total score: 2000 – ((1400 + 140) + 140) = 320

Again this option is also dependent on the users information need and will not be a weighting factor for all user queries.

4.6.5 Ostensive Object Weighting

As with the other features used in shot ranking, a user's relevance judgements on objects ought to have a reduced weighting as a user's search iterates over time. The following is a modified version of location score example that demonstrates such a decay or weighting reduction:

Ostensive scoring example:

Corpus shot ***x*** and RJ shot ***y*** are being compared. Scoring based on visual and Object-based features with a visual score of 3000 and object matching score of **1400**. Ranking is in ascending order. **Note**: visual shot scores are dissimilarity-based whereas object scores are similarity-based.

The visual score for shot_dissim$^{\text{visual}}$ (x,y) = 3000
Ostensive weighting: 3000 * 1.0 = 3000

x and **y** have 2 relevant matching objects with ostensive weightings of 0.9 and 0.8 while ***y*** contains **1** non-relevant matching object with an ostensive weight of 0.7.

1400 * 0.9 = 1260 (Relevant object 1)
1400 * 0.8 = 1120 (Relevant object 2)
1400 * 0.7 = 980 (Non-relevant object 1)

The Total score: ((3000 – 1260) – 1120) + 980 = **1600**

4.7 Summary

In this chapter we have introduced and described an ostensive Relevance Feedback model for video shot retrieval that incorporates text, low-level visual features and template-based object information within the formula used for ranking and retrieval of video shots. The model uses video shots as the unit of retrieval and extracted feature information is aligned to this structure. Two methods of feature value comparison were discussed, absolute difference and intersection, where the former is the difference between two values while the later is the value they have in common.

Following the introduction of the model, we then discussed how the model works with text where the first step is applying stemming and stopword removal techniques. The remaining text is then aligned at the shot level with TF-IDF (a standard text-based information retrieval technique) applied to it. A 'value' is given to each term (word) match until a total score is obtained. The final step is to reduce the similarity score based on the age of the shot RJ using an approach known as ostensive weighting.

Following on from including text in shot retrieval, we then discussed how low-level visual features are used in the model. Unlike text, the visual features discussed in this chapter were not run on each frame of the complete shot as there would be too much computational expense and redundancy and storage requirements would be excessive. Instead, a single representative 'keyframe' is used with visual features extracted from this. While this is a short-cut it allows us to implement our model with reasonable overhead. Shot comparison scores for each of the features requires normalisation with a ratio-averaging technique before their combination. Having obtained a shot dissimilarity score the weight is increased based on the age of the shot RJ.

Finally we discussed how the model could use template-based object information in the shot ranking process. A score is given if matching objects are present in the shot RJ and the evaluation shot, and this score is increased for a positive judgement and reduced for a matching negative judgement. The facility also exists for similar sized objects and matches found in the same location to also effect the final object comparison score. Having obtained a shot similarity score the result is decreased based on the age of the shot RJ, where "age" is determined as a function of the sequence of shots judged by the user.

The text and object scores are similarity-based whereas the visual score is dissimilarity-based, and what this means in practice is that the complete text and object comparison scores are subtracted from the visual score in order to obtain the final shot comparison score.

In the next chapter we will discuss a video IR application based on this model and discuss a number of user experiments to evaluate ostensive object retrieval.

Chapter 5 The Físchlár Símpsons Experiment

5.1 Introduction

The previous chapter described a model of video retrieval that incorporated ostensive relevance feedback and object information. This chapter is organised in two main parts. The first part describes the *Físchlár-Simpsons* video retrieval application which incorporates the ostensive relevance feedback retrieval model and object information. The animated television show "The Simpsons" provides the video content that will be used in the experiment. The *Físchlár-Simpsons* graphical user interface (GUI) is discussed along with descriptions of the video content and search features.

The second part of the chapter discusses a video retrieval experiment using variations of the *Físchlár-Simpsons* application to probe the potential benefit of ostensive RF.

5.1.1 What type of content is The Simpsons?

The television show "The Simpsons" has been a phenomenon since it first started broadcast on US television in 1990. With over 340 episodes completed in 15 years, the show remains as popular as ever. The 340 episodes of the Simpsons are 136 hours or 5.6 days worth of content. The show is currently in its 16th season* which is something of a record for an animated show.

The basic premise of the show revolves around the exploits of a 'typical' dysfunctional American family "The Simpsons" and their hometown of Springfield.

Over the years, the show has evolved both in terms of the quality of the animation and the storylines. The animation quality for the first year of the show was poor and was greatly refined for the second season. In the following years the animation was further refined and polished but the largest change appears from the first to second year. Homer's voice and characterisations changed greatly after the first season.

In terms of storylines the first two years of the show tended to focus on Bart as the main character but towards the second year this focus shifted to Homer who became (and remains) the most popular character.

5.1.2 The Experimental Corpus

The *Físchlár-Simpsons* video retrieval application, as the name suggests, facilitates browse and retrieval on Simpson's content. In Chapter 3 we discussed object extraction explaining that animated content is the best choice for evaluation due to the nature of the content i.e. a limited number of specific colours with generally well defined object boundaries. As we explained in the previous section, "The Simpsons" is one of the most popular animated shows in the world with over 340 episodes (136 hours of content). For these reasons it was decided to select "The Simpsons" as the content for retrieval evaluation.

IR performance evaluation requires a representative selection of content and in the case of the *Físchlár-Simpsons* experiment it needs a selection of representative episodes. The general procedure for creating an experimental corpus is to select a quantity of representative content and divide it into two segments, a development corpus and evaluation corpus. The development corpus is used to gauge the performance of the retrieval features and system during the development cycle, while the evaluation corpus is used for the actual experiments and results. The main reason for using a separate corpus for development and for evaluation is to avoid training on a specific corpus and producing artificially high results.

[*] Generally, a season on American television is one year and contains 20 – 26 episodes depending on the programme whereas on British television a season generally consists of 6-9 episodes.

The first task for building a corpus is obtaining the content and in the case of the Simpsons a large quantity of video content is available on DVD and VHS making it easily available. Content on DVD is the medium of choice due to the availability of closed captions (subtitles) and the video's high quality.

At the time writing, three seasons of content was available along with a number of special themed collections*. As we discussed in the previous section, the first season of the Simpson's feature a different animation style to later seasons and for this reason it was not considered for inclusion in our corpus.

Table 5-1 shows all **12** episodes (**4.8** hours, **6,525** shots) included in the development corpus, episode number is the order used by the system while broadcast number is the actual order of the episode.

The two main considerations for inclusion of content in the development corpus are that the episodes contain appropriate training examples for the object detection (discussed in Section 5.1.3) and secondly that the corpus contains an appropriate quantity of episodes for low-level feature and system evaluation.

Table 5-1: Development Corpus Content

***Episode* Number**	**Episode Name**	**Broadcast Number**	**Season**
1	Bart gets an F	14	2
2	Simpson's and Delilah	15	2
3	Bart V's Thanksgiving	20	2
4	Bart the Daredevil	21	2
5	Itchy & Scratchy & Marge	22	2
6	Bart gets hit by a car	23	2
7	One fish, two fish, blowfish, blue fish	24	2
8	The way we was	25	2
9	Principal Charming	27	2
10	Brush with Greatness	31	2
11	Lisa's Pony	43	3
12	Separate Vocations	53	3

* The Themes featured four similar episodes from throughout the seasons and included collections like Bart at War, Halloween Specials and Simpsons go to Hollywood

Table 5-2 shows the **52** episodes (**20.8** hours, **20,529** shots) that make up the complete evaluation corpus. While most of the content comes from the 2nd and 3rd seasons the episodes from 13 to 29 were obtained from special themed DVDs which came from seasons 4 to 12.

Table 5-2: Evaluation Corpus Content

Episode **Number**	**Episode Name**	**Broadcast Number**	**Season**
1	Two cars in every garage and three eyes on every fish	17	2
2	Dancin Homer	18	2
3	Dead putting society	19	2
4	Tree House of horror	16	2
5	Homer Vs. Lisa and the 8th Commandment	26	2
6	Oh Brother, Where Art Thou?	28	2
7	Bart's Dog Gets an F	29	2
8	Old Money	30	2
9	The Substitute	32	2
10	The war of the Simpsons	33	2
11	Three men and a comic book	34	2
12	Blood Feud	35	2
13	Simpsons Halloween 5	86	5
14	Simpsons Halloween 6	109	6
15	Simpsons Halloween 7	134	7
16	Simpsons Halloween 12	249	12
17	When you dish upon a star	208	10
18	Fear of Flying	114	6
19	Krusty gets Kancelled	81	4
20	Tree House 9	182	9
21	The Cartridge Family	183	9
22	Natural Born Kissers	203	9
23	Grandpa Vs Sexual Inadequacy	113	6

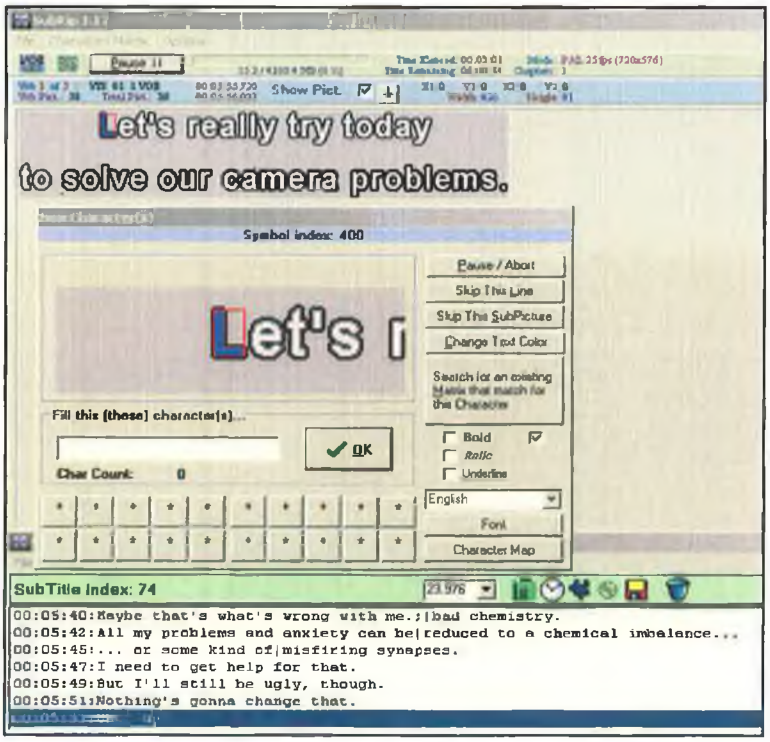
24	Mayored To The Mob	212	10
25	The Secret War Of Lisa Simpson	178	8
26	Marge Be Not Proud	139	7
27	Homer To The Max	216	10
28	The Springfield Files	163	8
29	Lisa The Iconoclast	144	7
30	Homer Badman	112	6
31	Stark Raving Dad	36	3
32	Mr. Lisa Goes To Washington	37	3
33	When Flanders Failed	38	3
34	Bart The Murderer	39	3
35	Homer Defined	40	3
36	Like Father Like Clown	41	3
37	Treehouse Of Horror 2	42	3
38	Saturdays Of Thunder	44	3
39	Flaming Moe's	45	3
40	Burns Verkaufen Der Kraftwerk	46	3
41	I Married Marge	47	3
42	Radio Bart	48	3
43	Lisa The Greek	49	3
44	Homer Alone	50	3
45	Bart The Lover	51	3
46	Homer At The Bat	52	3
47	Dog Of Death	54	3
48	Colonel Homer	55	3
49	Black Widower	56	3
50	The Otto Show	57	3
51	Bart's Friend Falls In Love	58	3
52	Brother, Can You Spare Two Dimes	59	3

The Simpson's episodes used in the corpus are stored on DVD in high quality MPEG-2 video format and this needs to be extracted and converted into MPEG-1 for image analysis, storage and playback.

The first step requires the extraction of the video data from the DVD onto the computer hard drive, from here the second step converts each episode to MPEG-1 using a transcoding application like Flaskmpeg [Flask].

Extraction of the closed captions presents a more difficult challenge, as the DVD subtitle text is actually stored as a series of images (white text on a black background). Therefore, optical character recognition (OCR) is required to convert the image text into machine-readable text.

Figure 5-1 shows a screenshot from "subrip" a DVD subtitle extraction application which takes the video data and converts the subtitles into a structured machine readable text file. It displays characters that it cannot identify with a red box, and as it is shown the characters it learns and quickly works automatically. The extracted subtitle text and timing information is displayed of at the bottom of the screen and this can be saved to a file.



```
00:05:40:Maybe that's what's wrong with me.;|bad chemistry.
00:05:42:All my problems and anxiety can be|reduced to a chemical imbalance...
00:05:45:... or some kind of|misfiring synapses.
00:05:47:I need to get help for that.
00:05:49:But I'll still be ugly, though.
00:05:51:Nothing's gonna change that.
```

Figure 5-1: Screenshot of DVD subtitle extraction application

5.1.3 Físchlár-Simpsons Object Retrieval

As we have previously discussed *Físchlár-Simpsons* facilitates object retrieval and consequently a corpus of animated content was constructed with this in mind. The object detection feature used in the application is based on shape template matching and colour. Animated content tends to use a very specific palate of colours and characters with a fixed number of identifiable poses therefore it is feasible to identify selected objects automatically with a good degree of accuracy.

The Simpsons character object detection works by having a number of previously constructed templates for each character, extracting all yellow coloured objects no matter what they are, from a shot keyframe and then matching these yellow objects against character templates.

The procedure for detection training is straightforward with a number of representative examples (templates) for each character/object pose manually selected from representative images. The templates can then be used in the detection process to obtain similarity-matching scores.

In the detection process each template is compared against shape boundaries in the comparison image using a shape matching algorithm application which generates a number of comparison scores and any scores above a pre-defined threshold are regarded as a positive match.

Colour plays an important part in the shape matching process, as positive matches require yellow areas* matching the templates. All other colours are ignored and this reduces false detection considerably. [Adamek and O'Connor, 2003] [O'Connor et al., 2003]

Figure 5-2 is a screengrab taken from Adamek's template-based shape matching application [Adamek and O'Connor, 2003]. At the bottom of the screen, we can see the 5 members of Simpsons family and their representative templates. At the top left of the screen we can see the keyframe image that is being compared while the top right shows positive template matches for Homer, Lisa and Marge.

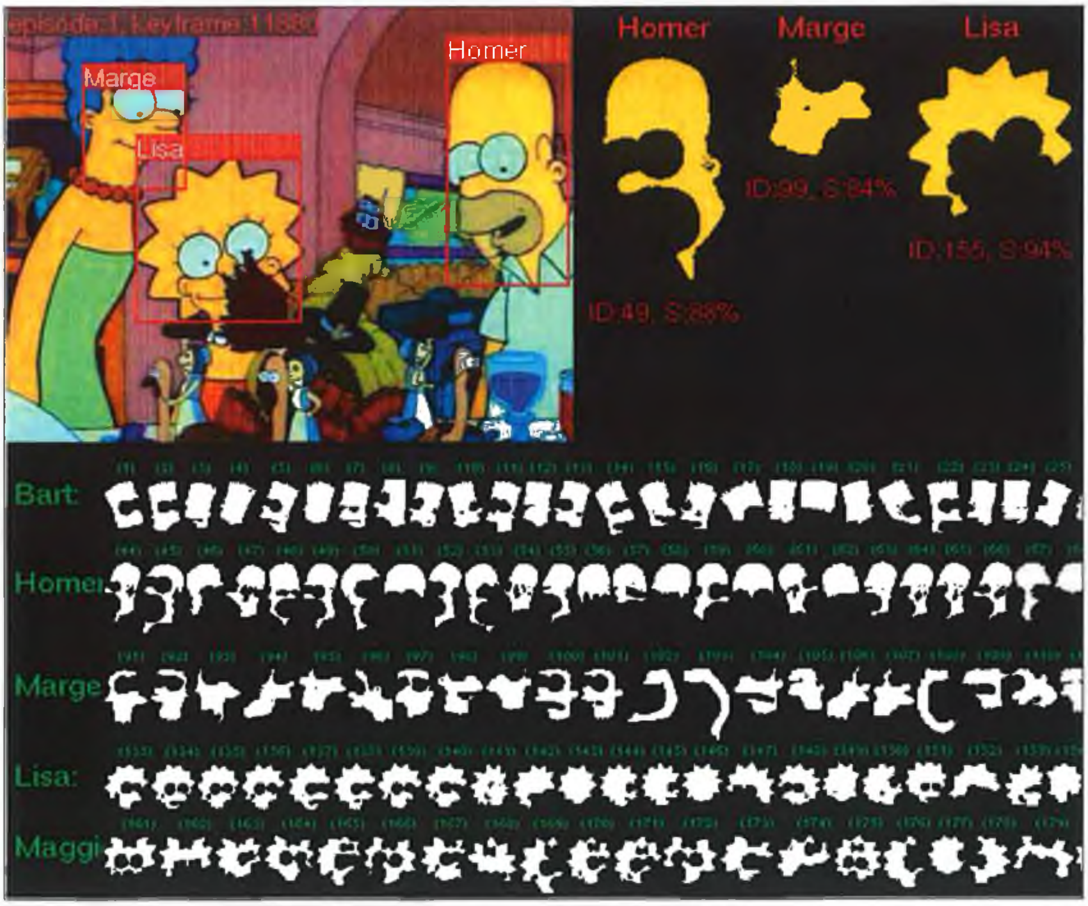


Figure 5-2: Screenshot of template matching application

The difficulty with the template matching technique is that it needs multiple training examples to detect each character, and this would make the approach impractical for detection of all characters. However, it is feasible to detect a small number of the more prominent characters.

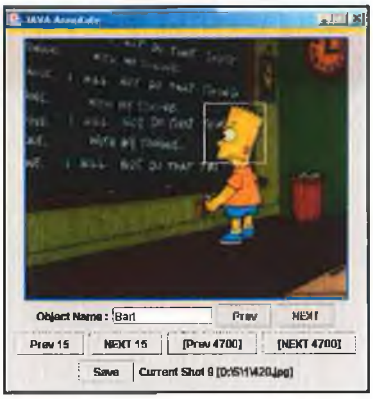
The decision of which characters to detect automatically is based primarily on their popularity and secondly on the availability of representative examples in the development corpus. Table 5-3 gives a list of the 10 characters that are detected automatically with this approach, their percentage occurrence in the development corpus (**P**), the number of shots where they occurred (**N**), and the accuracy of detection.

* In general, 'human' characters in the Simpsons have a yellow colour.

Table 5-3: Objects Detected Automatically in the Simpsons

Simpsons Character	Object Statistics	Simpsons Character	Object Statistics
1: Homer 	P: 32% N: 2066 Recall: 0.36 Precision: 0.98	**2: Marge** 	P: 21% N: 1343 Recall: 0.32 Precision: 0.87
3: Bart 	P: 21% N: 1361 Recall: 0.49 Precision: 0.98	**4: Lisa** 	P: 10% N: 647 Recall: 0.57 Precision: 0.93
5: Maggie 	P: 5% N: 321 Recall: 0.31 Precision: 0.88	**6: Burns** 	P: 4 % N: 269 Recall: 0.47 Precision: 1.0
7: Mr Smithers 	P: 2.6% N: 175 Recall: 0.46 Precision: 0.98	**8: Abe Simpson** 	P: 1.3 % N: 83 Recall: 0.27 Precision: 0.91
9: Principle Skinner 	P: 4 % N: 246 Recall: 0.8 Precision: 1.0	**10: Moe (bartender)** 	P: 0.7 % N: 43 Recall: 0.61 Precision: 0.78

The accuracy of object detection is calculated by creating a benchmark (ground truth) for each of the ten objects on the development corpus. The creation of the benchmark required each shot to be viewed with relevant results logged. Apart from the object name and the shot number the co-ordinates of the object also need to be output. Figure 5-3 shows a screenshot from the annotation programme that was developed to annotate objects on the development corpus.



Object Name : Bart

Current Shot 9 [D:\5\1\420.jpg]

Figure 5-3: Screenshot from the Benchmark Annotation Program

5.1.4 Físchlár-Simpsons Additional Features

Text search is available on the extracted DVD subtitle text. Each term in the extracted text is associated with a shot based on the timing information. Stemming and stopword removal are then applied to the extracted text, and the remaining terms are indexed based on the standard TF*IDF method. Text comparison between a given text query and shots is based on the vector space retrieval model and the resulting comparison scores are ranked in descending order.

Físchlár-Simpsons uses the following four low-level visual features (described in Section 2.2.3):

- 4 region * 18 bin hue histogram
- 9 region * average colour
- 9 region * median colour
- 4 region * 16 bin edge histogram.

Low level visual features are used to index each shot keyframe and facilitate two different approaches for search initialisation. The first approach is shot-based RF where a user clicks on shot(s) similar to the information need and retrieval is carried out based on the low level visual features detected from the chosen shot(s). The second approach is a drawing visualisation where the user draws the basic colours and/or shapes they are interested in and the drawing is converted into an image representation of the user's need, with retrieval done based on the low-level visual features of this representation.

In order to evaluate the performance of the system during development, 10 narrow query topics were chosen for retrieval of Simpsons content using the following four low level features:

- Regional Average Colour (9 Region * 3 Colour RGB)
- Regional Largest Colour (9 Region * 3 Colour RGB)
- Regional Hue Histogram (4 Region * 18 Colour bin)
- Regional Edge Histogram (4 Region * 16 Edge bins)

The reason for selection of narrow topics is that some might have valid results in over half the corpus of shots (a search for Homer for example). This part of the evaluation was developed to test system integration and the application of low-level visual feature extraction rather than object retrieval and is reported in detail in [Browne and Smeaton, 2004]. Details of this pre-experiment are not included here but can be found in [Browne and Smeaton, 2004]

Section 5.2.3 will describe two examples of retrieval on Físchlár Símpsons using text, low-level features and objects.

5.2 The Físchlár-Símpsons System

In this section we discuss the *Físchlár-Simpsons* application interface and show how it addresses the needs of an ostensive RF video retrieval system. There are a number of important requirements for the application, and they need to be implemented with a straightforward interface, as novice test users will be expected to use the application to perform retrieval experiments.

5.2.1 General System Requirements

The first functional requirement is for **relevance feedback**. Users should be able to provide multiple shots/ objects as relevance judgements of their information need and the system must re-rank the results accordingly.

The second functional requirement is the incorporation of the **ostensive** RF method of relevance feedback ranking. While the inclusion of ostensive RF is unseen by the user and does not effect the GUI, the system needs to be able to handle a number of ostensive weighting schemes.

The third aspect of functionality that is required is **text** retrieval, a user should be able to add text keywords from their query topic/ information need and receive a ranked list of shots based on their associated text.

The fourth functional requirement is for **low-level** feature search, a user should be able to search based on shots, visual colour and/or edge, shape and receive a ranked/ re-ranked list of results.

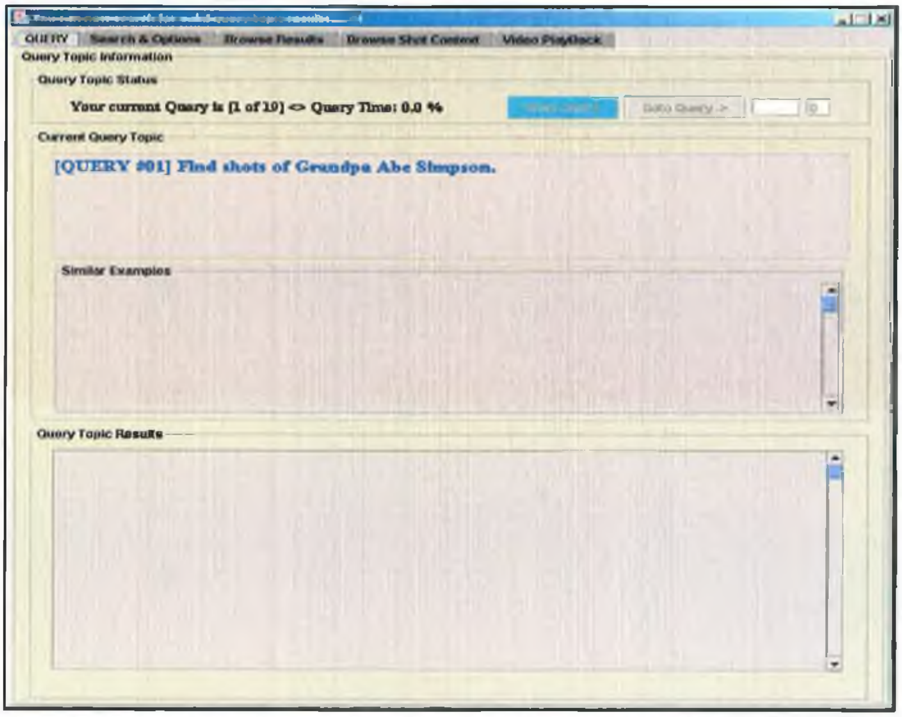
The fifth functional requirement is for **object search**. As we have seen in Section 5.1.3, object information is available for 10 characters and the system must provide the facility for ranking/ re-ranking of shots (RF) based on users' object selections.

The final functional requirement is the ability to view the **context** of a ranked shot result. In many cases a query topic might have relevant answers near the same location of a ranked result, therefore it is important to be able to view the shot context (shots that came before and after a ranked result)

5.2.2 GUI System Interface

The *Físchlár-Simpsons* application is broken down into five main sections: query topic, query options, browse shots, shot context and playback.

1. The **Query** screen (Figure 5-4) is used during the experiment and shows the user each query topic in order and any result shots saved as results. A user is given a set amount of time for each query topic after which they are automatically returned to this screen and shown the next topic.



Your current Query is [1 of 10] <> Query Time: 0.0 %

[QUERY #01] Find shots of Grandpa Abe Simpson.

Figure 5-4: Screenshot of the Query Screen from Físchlár Simpsons

2. The **Search & Options** screen (Figure 5-5) provides users with the facility to do a text search and draw a visualisation of their information need. Text and drawing searches are primarily used to generate the initial shot ranking, query refinement and relevance feedback is accomplished using the browse results and browse shot context screens.

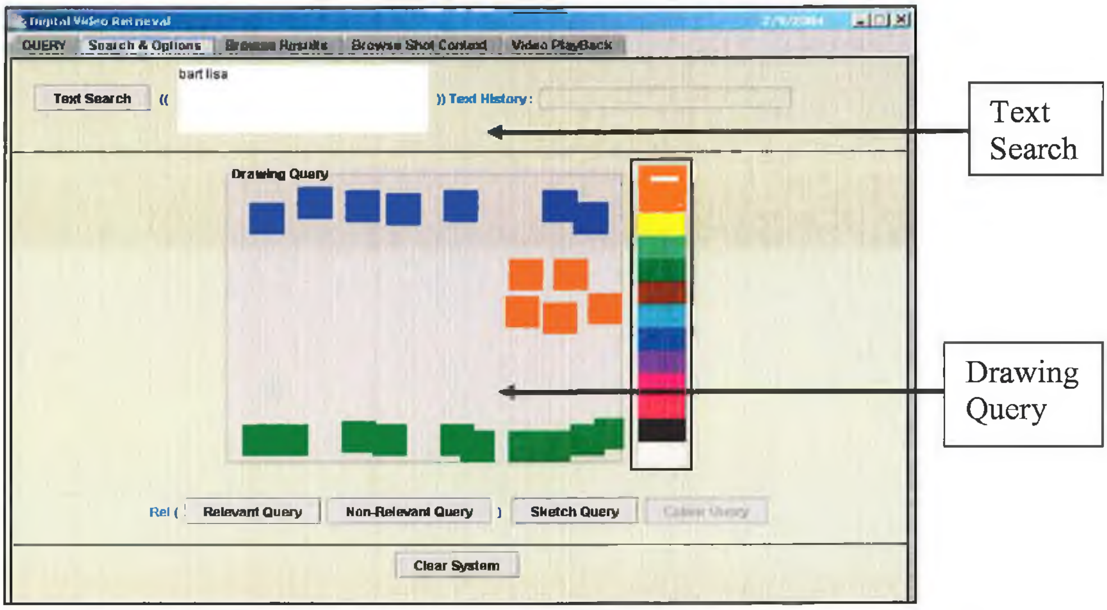


Figure 5-5: Screenshot of the Search & Options Screen from Físchlár Simpsons

3. The **Browse Shots** screen (Figure 5-6) shows the ranked shot results from the users query interaction. Relevant and non-relevant shots (low-level features) are shown at the bottom left while relevant and non-relevant objects are shown at the bottom right. Ranked shots can be added as either positive or negative feedback (RF) from this screen and when a judgement is made the results displayed here are automatically updated.

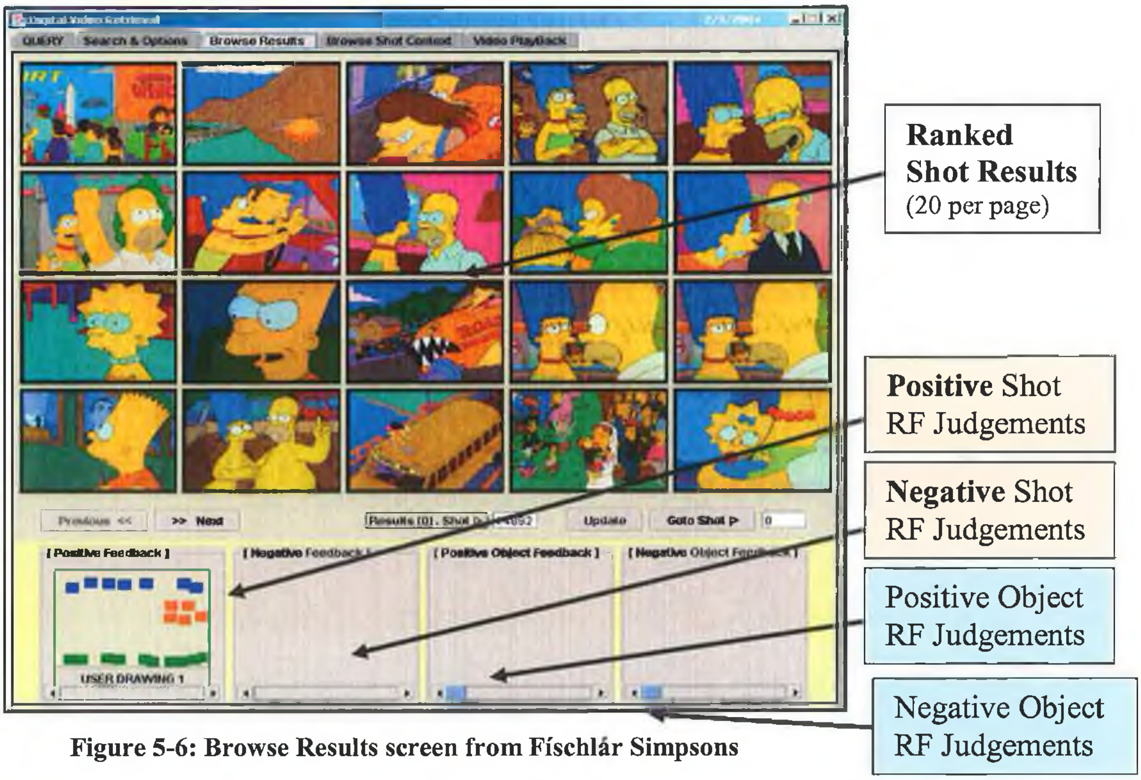
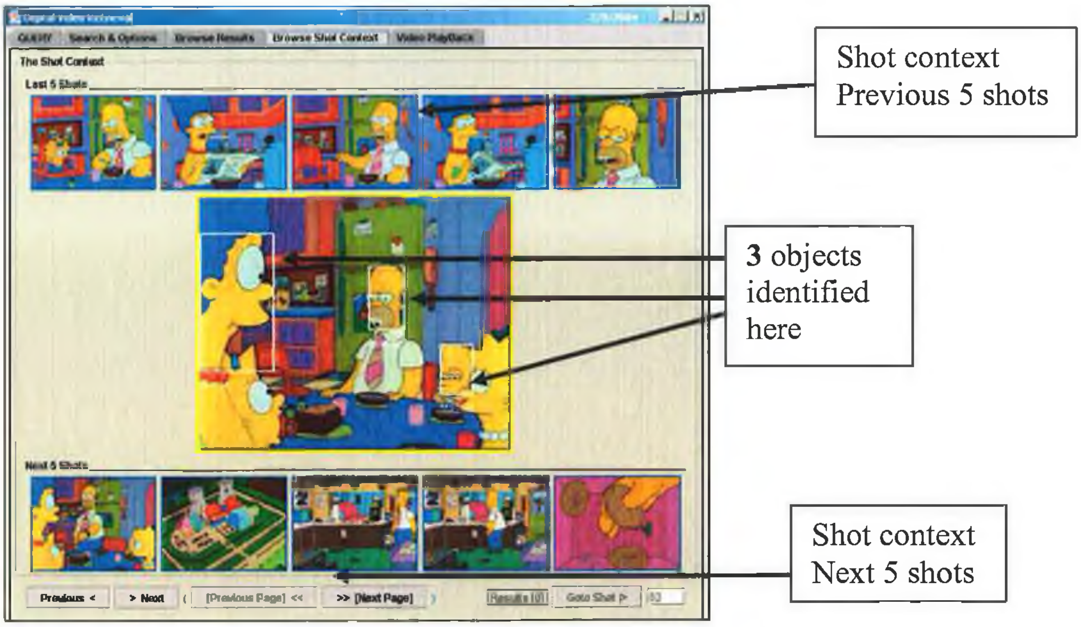

Digital Video Retrieval
QUERY
Search & Options
Browse Results
Browse Shot Context
Video PlayBack
Previous <<
>> Next
Results [0] . Shot
Update
Goto Shot ▷
0
[Positive Feedback]
USER DRAWING 1
[Negative Feedback]
[Positive Object Feedback]
[Negative Object Feedback]
Ranked Shot Results (20 per page)
Positive Shot RF Judgements
Negative Shot RF Judgements
Positive Object RF Judgements
Negative Object RF Judgements

Figure 5-6: Browse Results screen from Físchlár Simpsons

4. The **Shot Context** screen (Figure 5-7) shows the context (previous 5 shots and next 5 shots) of a given ranked shot result shown in the centre of the screen and allows the addition of object judgements to the users query. Objects are displayed on this screen with a white boarder and can be added as positive and negative RF by clicking within the white border.



Shot context Previous 5 shots

3 objects identified here

Shot context Next 5 shots

Figure 5-7: Browse Shot Context screen from *Físchlár-Simpsons*

5. The **Playback** screen shows the Simpsons video, to start playback the user selects any shot from the result or context screen with playback start time based on the location of the shot within the video.

Interaction with the system requires a combination of the computer mouse and keyboard.

Mouse Commands

- Left-click the mouse on a shot to add it as relevant to the query.
- Left-click the mouse on an object within a shot to add the object as relevant to your query.
- Right click the mouse on a shot to mark it as non-relevant to your query.
    - Right-click the mouse on an object within a shot as non-relevant to your query.

Keyboard Commands

- **F1** to **F5** takes you to each of the five main screens.
- **P** (Mouse over a shot) starts playback of video at that point.
- **R** (Mouse on a shot) adds this shot as a relevant query result.
- **C** cancels the query judgements but does not remove any results added.
- **V** (Mouse on a shot) displays the context of the shot showing the 5 shots that occur before and after it.

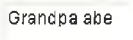
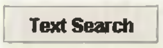
5.2.3 Examples of User-based RF interaction

In this section we will look at two query topic examples and discuss how the *Físchlár-Simpsons* video retrieval application could be used in order to find results.

Query Topic Example One: Find shots of Grandpa 'Abe' Simpson.

The first task for the user is to read the topic from the query screen and understand its requirements. **Note,** as one of the available objects is 'Abe', object search **will** help for this topic.

- When he/she is ready, the user can click on the button to start their search.

- The next task is how to find an 'Abe' object, so the user adds the following terms in the search & options screen for a text search  and clicks 

- The shots are ranked and displayed to the user on the browse results screen (Figure 5-8):

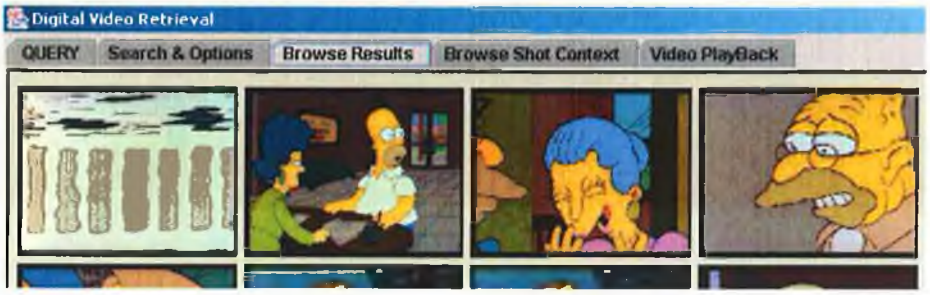


Figure 5-8: Top 4 Ranked Results Topic One first iteration

- There is one relevant shot on the right and this can be added as a result by moving the mouse over them and clicking '**r**'.
- Left click the mouse over the fourth ranked result shown in Figure 5-8 to indicate it as relevant (Figure 5-9).

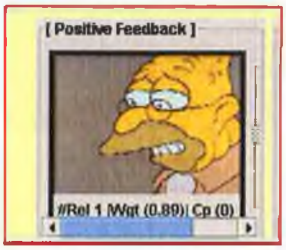


Figure 5-9: Positive RF Judgement

- Results are re-ranked for the second time (Figure 5.10).

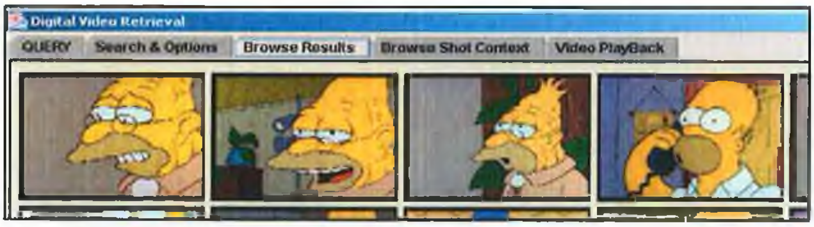


Figure 5-10: Top 4 Ranked Results Topic One Second iteration

- Add the second and third ranked results as relevant by moving the mouse over them and clicking '**r**'.
- Move the mouse over the third ranked result and click '**v**' to view the shot context (Figure 5-11).

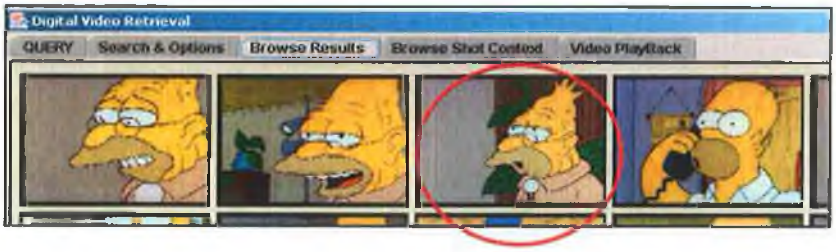


Figure 5-11: Select Shot Context

- The context screen shown in Figure 5-12 displays a number of relevant results that can also be added by moving the mouse over them and clicking '**r**'.

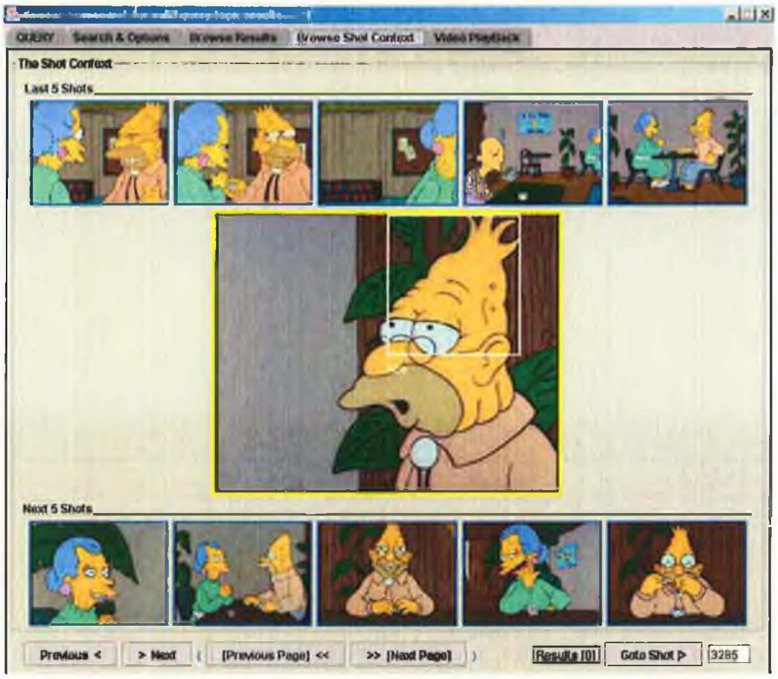


Figure 5-12: Context Screen Topic One

- Select the 'Abe' object shown in Figure 5-12 as relevant (Figure 5-13)

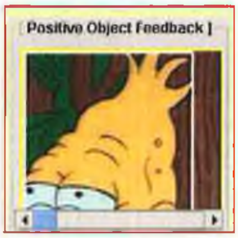


Figure 5-13: Positive object detection Topic One

- Figure 5-14 shows the top 4 results from the third re-ranking

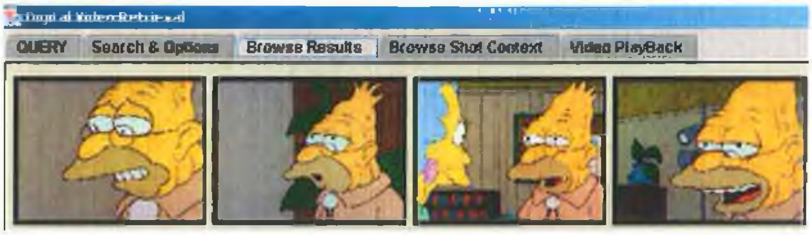


Figure 5-14: Top 4 Ranked Results Topic One, third iteration

Query Topic Example 2: Find shots of any of the two Simpsons cars.

The first task for the user is to read the topic from the query screen and understand its requirements. *Note:* none of the objects available is a 'car', so object search **will not** help for this topic.

- Having read the topic click on the Start Query button to start the search.

- From the Search & Options screen a visual representation of the information need is drawn (Figure 5-15). The pink colour represents the car while the blue at the top of the image is used to represent blue sky.

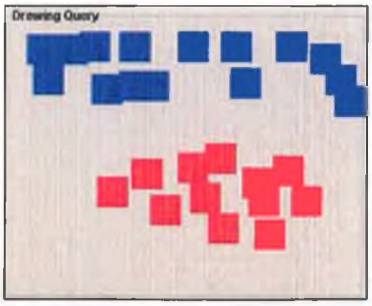


Figure 5-15: User Drawing Query Topic Two

- When the drawing is completed click on Relevant Query to rank results for display.

- Two relevant results are shown in Figure 5-16, move the mouse over them and click '**r**' to add the result.

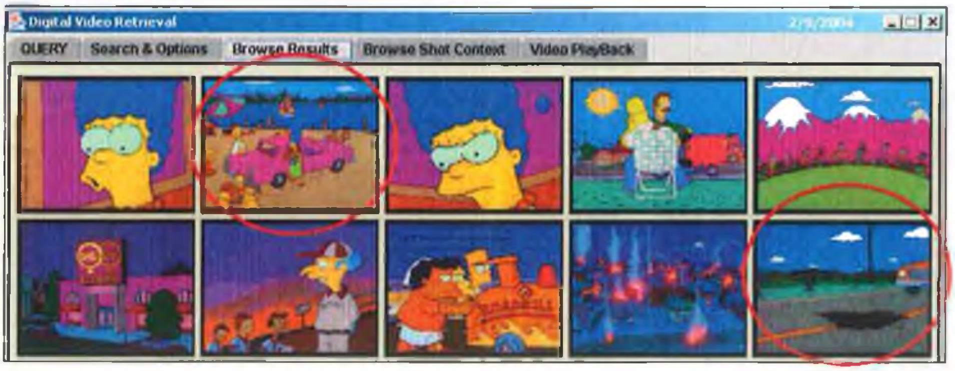


Figure 5-16: Top 10 results for topic two, first iteration

- The second result (shown in Figure 5-16) could help find more relevant results, move the mouse over the shot and left-click to add the shot as a relevant judgement.

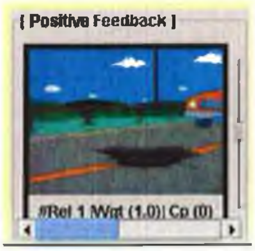


Figure 5-17: Topic Two Relevance Judgement

- As we can see from Figure 5-18 there are a number of relevant results that have been found.

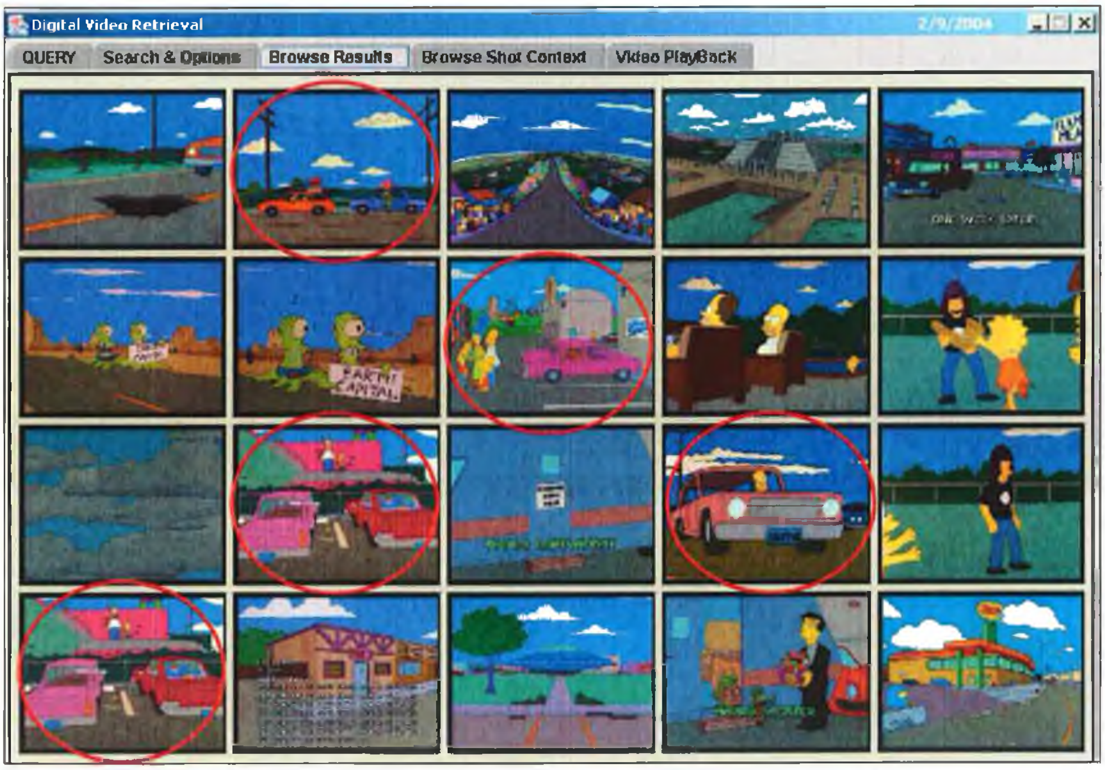


Figure 5-18: Top 10 results for topic two, second iteration

5.3 Evaluation Experiment Setup

5.3.1 Experiment Setup Introduction

In this section, we will discuss the video retrieval experiment in more detail. The first part will focus on the query topics that the experiment will use, the second part will discuss at the variation in evaluation systems, the third part will describe the experiment details, and the final part will discuss the creation of a corpus benchmark.

5.3.2 Experimental Query Topics

As we have seen in Section 5.2.2, the *Físchlár-Simpsons* application has a query topic screen that displays to the user the query topic and any associated results found. The process of query topic selection is a challenge as it can profoundly effect the performance results obtained by the user. An experiment testing the performance benefit of object retrieval and few query topics requiring the use of objects would show little benefit for object inclusion when the overall results are studied. A majority of query topics that require objects would very likely show the opposite result.

Query Topics can differ in terms of recall, where very broad topics will have many relevant results possible whereas narrow topics might only have a hand full of possible results. If we wish to evaluate two systems and give one the broad topic and the other the narrow topic then very likely the system with the broad topic will perform better regardless of the quality of the two systems.

Query topics can also differ in terms of retrieval complexity with some requiring several steps of query search from the user before any relevant results can be found whereas others might only require the addition of a few text terms before finding numerous relevant answers. While balancing all the requirements for query topics is not always possible, the evaluation experiment must consider them all carefully.

Query topics are selected in two groups, the first group is needed for user training and should require them to use the various aspects of the evaluation system, and the second group is used for the evaluation experiment from which the results will be extracted.

Table 5-4 shows the five training topics used in the experiment and have been chosen for the following reasons:

1. The first topic tests the user's ability to do a basic text and object search, a text search will find 'Abe' and when followed by an object search will find most of the available results.
2. The second topic requires a visualisation, text search and low-level features in order to find relevant results.
3. The third topic requires a text search and shot context to find the relevant results. The context is important in this topic as all relevant results occur in only one episode.
4. The fourth query topic requires relevant and non-relevant objects in order to find relevant results.
5. The final training topic requires a combination of text, object and low-level shots in order to find relevant results.

Table 5-4: Training Query Topics

Query Topic	Topic Description
1	Find shots of Grandpa Abe Simpson.
2	Find shots of the Simpsons' cars. A relevant result features any of the two Simpson cars.
3	Find shots of video that show Krusty the clown's father Rabbi Krustofski
4	Find shots of just Maggie and Marge. Relevant results will show just these two characters.
5	Find shots of Homer drinking beer in Moe's Tavern. Relevant results should show Homer with and/or next to a glass of beer.

Table 5-5 shows the evaluation query topics from which performance results will be extracted and compiled. The topics require search using: text (T), low-level feature (L), object (O), drawing (D), context search (X) and combinations of features (C), as we can see from the search strategy column.

Table 5-5: Evaluation Query Topics

Query Topic	Topic Description	Search Strategy
1	Find shots of any Simpson character smoking. Shots where the character is holding a cigarette is also relevant.	(L), (T)
2	Find shots of people hurt and in pain. The sound must be audible.	(T), (X)
3	Find shots of Bart, Lisa and Maggie together in the sitting room WITHOUT Homer. Shots can contain other characters but must contain Bart, Lisa and Maggie and NOT homer.	(O), (L), (T)
4	Find shots of Homer working at the nuclear power plant. Relevant results can show him at his desk or 'working' around the plant. Shots in the plant canteen are also relevant.	(T), (L), (O)
5	Find shots of Bart Simpson sad, upset, worried. The emotion should be evident on his face and/or in his voice.	(O), (X)
6	Find shots that show Bart and a teacher. Relevant results will show JUST Bart and a teacher, no other character should be visible.	(O)
7	Find shots that show an explosion. The sound must be audible.	(D), (X), (T)
8	Find shots that show Homer's neighbour Ned Flanders.	(T), (L), (X)
9	Find shots of the 'new' owners of the Springfield nuclear power plant when Mr Burns decided to sell. Note: Shots of Mr Burns alone are not relevant. Relevant results are any of the new owners.	(T), (X)
10	Find shots of the Simpsons character that proposes to Marge's sister Selma. While a number of Simpson characters proposed to Selma over the years there is only ONE correct answer in this content.	(T), (X), (L)
11	Find shots of video that feature any sporting event. A relevant result can show a sporting event taking place and/or being watched by any of the Simpson characters.	(T), (L)
12	Find shots of Homer and Marge in the Kitchen without Bart, Lisa and Maggie.	(O), (L)

5.3.3 Evaluation Systems

In the previous section the query topics were described, the next important task is to decide what system variations to evaluate. In the case of *Físchlár-Simpsons* an ostensive relevance feedback model is being evaluated and therefore requires at least two system variations to be created, the first system has RF without ostensive weighting while the second incorporates ostensive weighting. In Chapter 3.2.2 two possible ostensive weighting curves were discussed, a linear decay and a sharper logarithmic decay, with this in mind two ostensive systems, each with a variation in decay, will be evaluated giving 3 systems altogether.

Figure 5-19 shows the shot decay rate of the **three** evaluation systems over 10 iterations*. The non-ostensive (**sys_non**) has a constant weight of 1.0, linear ostensive (**sys_ost1**) has a gradual decay while sharp ostensive (**sys_ost2**) has a type of logarithmic decay with a faster decay rate for initial iterations.

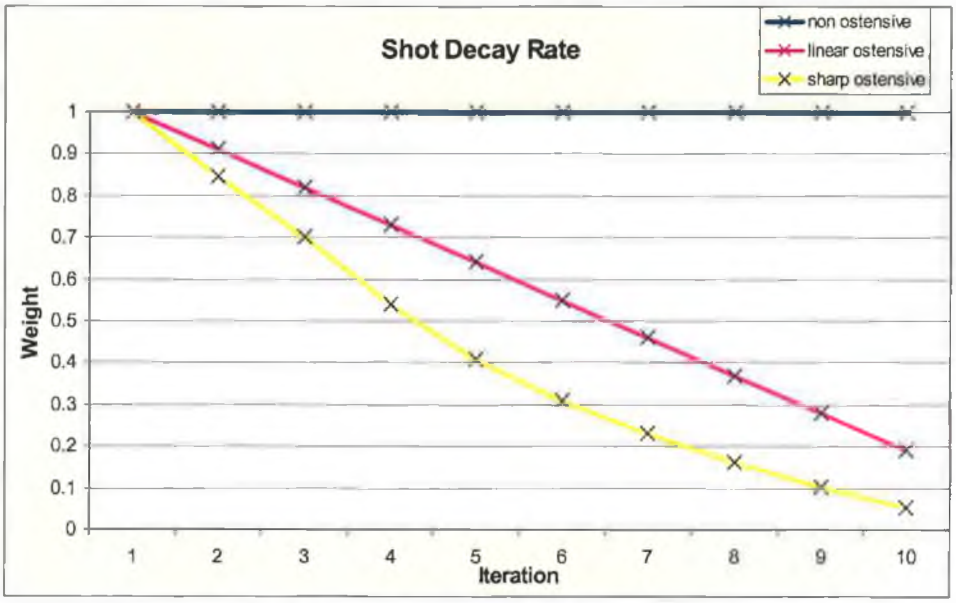


Figure 5-19: Shot decay weights over 10 iterations

While the decay weight of RF judgements is different for the **three** systems the actual interface for each is **exactly the same** with any system variation completely transparent to the user. The same interface will make it easier to compare the system variations as the search time taken and behaviour should remain the same.

5.3.4 Experiment Details

We have previously discussed twelve query topics will be completed by each user during the experiment and the three system variations will also be evaluated. In this section we will discuss the details of the experiment itself.

Experiment Time

The first task is to decide on the correct amount of **time** to allocate for each query topic. There is no standard amount of time to choose however, too little time (2-3 minutes) and the user might fail to find any relevant results, too much time (10-15 minutes) and the user might get frustrated or tired. The time taken to train users on the system, completion of test queries and the reading of query topic text also need to be considered.

TRECVID evaluation in 2003 recommended a maximum time of *15* minutes per query for manual systems. Including training time this would take users nearly 4 hours to compete, this might be too much to expect from users and reduce the number of experiment sessions that could be run each day. A time of *7* minutes per query would reduce the experiment time to just over 2 hours and allow two sessions to be run each day, this is the time decided on for these experiments. The following is the calculation of the experiment time:

• Experiment Topics (12 * 7 minutes)	: 84 minutes
• Questionnaire Topic Finish (12 * 1 minute)	: 12 minutes
• Query Topic Preparation (12 * 1 minute)	: 12 minutes
• Training Time Topics (5 * 7 minutes)	: 35 minutes
Expected Duration (approx.)	: **143** minutes (2 hours 23 minutes)

[*] After 10 iterations the weight decay stops in order to allow the RF judgements to contribute to shot ranking.

When each topic is finished (after 7 minutes), the user is returned to the query screen on *Físchlár-Simpsons* and shown the next query topic, the time only starts when the user clicks the start query button.

User Training

User driven experiments generally need to give some element of training to users before meaningful data can be collected, and with this in mind the following is the approach that was used for the experiment:

- The supervisor[*] gave a system demonstration and experiment description to each group of participating users. The first 2 test queries were completed by the supervisor during which time possible retrieval techniques and search strategies discussed.

- Each user completed the three remaining test queries on their own under supervision. During the training, suggestions were made by the supervisor when the wrong approach was attempted. Mistakes that could be made by users include: not using objects when useful, not giving an RF shot judgement when similar results might help find relevant results or a failure to use text search.

A 'typical' training session would be expected to take 35 minutes to complete for each group of users.

Query Rotation

In order to reduce the impact of user fatigue on individual topic results the 12 query topics are applied in three possible rotations, **normal** displays the queries in ascending order (topics 1-12) to the user, **descending** displays the queries (topics 12-1) and **split** (topics 7-12 then 1 – 6).

[*] The supervisor is the author of this work.

System Rotation

In order to reduce the impact of individual user bias on system performance each of the three systems was used in rotation by the user for the 12 topics. After completing the experiment the user will have completed 4 topics on each of the three systems.

Apart from the system rotation the start system that each user has for their first topic was also varied thereby reducing the bias of individual topics. The following two examples illustrate both rotations:

System Rotation Example One:

- User X starts with topic **1** with system **1**

Then

- Topic 2 with system 2

Then

- Topic 3 with system 3

Then

- Topic 4 with system 1

....

System Rotation Example Two:

- User Y starts with topic **1** with system **2**

Then

- Topic 2 with system 3

Then

- Topic 3 with system 1

Then

- Topic 4 with system 2

....

Query and system rotation is discussed in more detail in Chapter 6.3

Questionnaire

When each user finished a query topic they were asked to fill in a short one-page topic questionnaire (Appendix Figure B-1). The purpose of this survey was to find out how the user solved the query topic and gauge their impression of the various features. The topic questionnaire and user feedback can be seen in the Appendix B.

When each user has completed the experiment they were given a final one-page questionnaire (Appendix Figure B-2) which gauged their general knowledge of the Simpsons as well as an overall impression of *Físchlár-Simpsons*. The final questionnaire and results can be seen in the appendix Tables B-1 and B-2. The results in appendix B-1 show that the average user knowledge of the Simpsons is quite high (average rating 6.6 out of 10) while the level of comfort users felt working on *Físchlár-Simpsons* was very high (8.1 out of 10).

5.3.5 Creating a Performance Benchmark

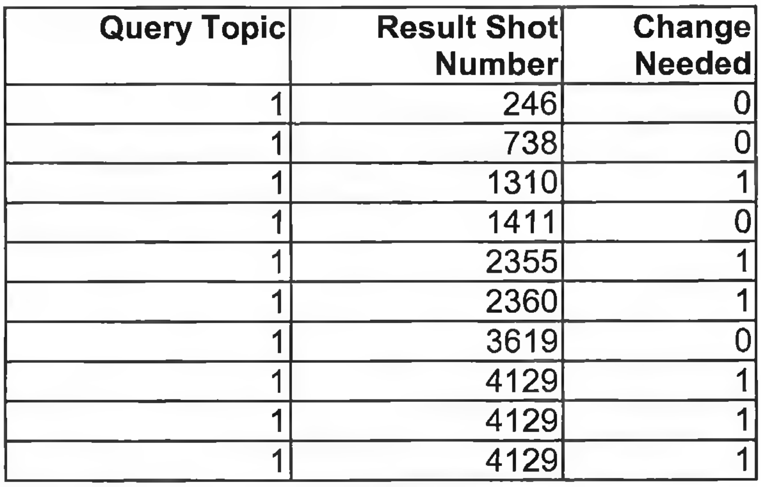
In order to obtain experimental results it is necessary to create a benchmark (ground truth) for the 12 evaluation query topics. The creation of the benchmark requires that all 20,529 shots in the evaluation corpus were evaluated for relevant answers on the 12 query topics with shot details logged to the appropriate query benchmark text file. When all 20,529 shots had been evaluated manually by the author and a postgraduate student the 12 complete query benchmark text files could be used to evaluate experimental performance.

Refining the Benchmark

In order to improve the accuracy of the manually created benchmark all the user results from the experiment were compared with the existing benchmark files and false results for the 12 query topics outputted to a text file. The text file was loaded into a spreadsheet application and the false results sorted in ascending order. Finally the *Físchlár-Simpsons* application was used by the author to manually evaluate each false result and, where required, the benchmark files were updated.

Table 5-6 is a sample of the false results with the change column used to register any updates required in the benchmark files, a '0' indicates no change while a '1' indicates a required update.

Table 5-6: Excel Benchmark Refinement Sample



Query Topic	Result Shot Number	Change Needed
1	246	0
1	738	0
1	1310	1
1	1411	0
1	2355	1
1	2360	1
1	3619	0
1	4129	1
1	4129	1
1	4129	1

The two-pass approach of creation and refinement reduces the effect of human error in the benchmark and the results from the experiment (discussed in Chapter 6) all use the refined benchmark files.

5.4 Summary

The start of this chapter discussed the television show "The Simpsons" describing the main characters and the evolution of the show over the years. The popularity of the show coupled with the need for animated content on which to run object detection explains why it was chosen for *Físchlár-Simpsons*.

In the next section the chapter discussed the experimental corpus explaining the need for a separate training and evaluation corpus and detailing the content in both. The process of corpus construction from DVD was discussed with the difficulty of subtitle extraction highlighted.

The chapter moved on to discuss the object retrieval feature used in *Físchlár-Simpsons*. A number of character templates were created for the 10 main Simpson characters with object matching based on the shape similarity and yellow colour. All Simpsons characters are yellow making object detection an easier prospect as similar matching templates with different colours can be ignored.

The next section of the chapter discussed the *Físchlár-Simpsons* application that will be used for the video retrieval experiments. The program is divided into 5 sections and facilitates ostensive RF-based retrieval using low-level features, text, drawing visualisation and objects.

The final part of the chapter focused on the setup for the evaluation experiment discussing the choice of query topics, their importance and the search strategy needed. Three *Físchlár-Simpsons* system variations will be tested, two have ostensive RF with a variation in judgement weight decay while the remaining system uses standard RF. For the experiment users are given training of 35 minutes on the system with 5 training topics before moving on to the actual experiment. 12 evaluation topics are given with query and system rotation used to reduce user and topic bias in the results.

In the next chapter we will discuss a number of popular IR performance evaluation measures before focusing on the results of the retrieval experiment described in this chapter.

Chapter 6 Retrieval Experiment Results

6.1 Introduction

In the previous chapter we discussed the *Físchlár-Simpsons* video retrieval application that incorporates ostensive-based relevance feedback for video retrieval and an experiment designed to evaluate retrieval performance.

In this chapter we will discuss the popular IR performance measures before looking at the profile of the users who provided the experimental results. The central part of the chapter focuses on the results from the experiment described in Chapter 5 which has been developed in order to probe the potential benefit of ostensive-based relevance feedback.

6.2 Retrieval Evaluation Measures

In this section we discuss the three popular IR performance evaluation measures. Evaluation measures are essential as they provide evidence of performance gain or loss and can help contrast one approach with another. The thesis is concerned with video retrieval and the unit of retrieval that will be evaluated is the shot, whereas for web-based IR evaluation the general unit for retrieval would be a document (html page). The following are the main terms that will be discussed in this section:

N_i: Number of shots found to be non-relevant for a given topic

N_c: Number of shots found to be relevant to a given topic

N_t: Actual number of relevant shots in the collection

6.2.1 Recall

Recall is used to measure the percentage of relevant results found and is calculated by dividing the number of returned relevant results N_c by the total number of relevant results N_t [Salton, 1983] [Van Rijsbergen, 1979]. The recall formula is as follows:

$$\text{Recall} = \frac{Nc}{Nt}$$

100% recall can be obtained if all relevant shots in the search collection have been found. Considering two different systems A and B, both systems achieve 100% recall, but system A needs to return 100 shots before it reaches 100% recall while system B needs to return 500 shots, therefore recall needs to be contrasted with precision before an accurate overall measure of performance can be obtained.

In some cases it is not possible to obtain an accurate measure of recall, for example large-scale web retrieval evaluation, as millions of documents would need to be evaluated. For TREC evaluation a pooling method is used where results from various participating groups are collected, duplicate results are removed, and the remainder are manually assessed by information specialists in order to create an evaluation baseline [Harman D, 1992]. It is hoped in this approach, first adopted in the Cranfield experiments of the 1960s, that by having a large and diverse set of retrieval results providing input into the pool the end result will be that almost all, or certainly *most* relevant documents, will be identified.

6.2.2 Precision

The precision measure is used to indicate the percentage of returned results that are relevant. Precision can be calculated by dividing the number of returned relevant results N_c by the total number of returned results (N_c and N_i, where N_i is the number of non-relevant results returned) [Salton, 1983] [Van Rijsbergen, 1979]. The Precision formula is as follows:

$$\text{Precision} = \frac{Nc}{Nc + Ni}$$

100% precision can be obtained when all shots **returned** to a given topic are relevant. For example, if only one shot is returned and it is relevant then precision would be 100%, therefore it is important to look at this measure in conjunction with recall.

Precision and recall are measures usually calculated at some rank position and in retrieval systems such as web search engines which return a ranked list, precision and recall are calculated at many points throughout the ranking and then precision values are mapped to a set of standard recall points of 0.0, 0.1, 0.2, 0.3 etc. A graph of precision verses recall is often used in IR to contrast the two measures and to illustrate performance throughout the document ranking.

6.2.3 Mean Average Precision (MAP)

The mean average precision (MAP) is essentially the mean of the average precision over **N** evaluation topics and applies to retrieval strategies which have a single definite cutoff point in their ranking rather than a ranking of the entire document collection. Before the mean average precision can be calculated, the average precision is computed for each evaluation topic. When all topic averages (TP) have been computed a MAP is calculated on the sum of all TP scores divided by the number of topics **N**.

An average topic precision **(TP)** is calculated by taking each relevant result and the position from which it was returned and calculating the precision at that position, averaging the precision over all relevant results. Each **TP** score is stored and used to compute the MAP score. An alternative way to calculate MAP for retrieval strategies which do not have a single definite cutoff point in their ranking is to calculate precision at the 11 standard Recall points of 0.0, 0.1, etc. and to average them. MAP has become popular as a measure because it is a single figure which is always desirable in comparing across systems.

6.3 The Experiment and User Details

The following section is a general profile of the users who provided the experimental results and the main experiment details.

- The number of test users was **15**.
- **Each** user was a postgraduate student and member of the Centre for Digital Video Processing [CDVP].
- **Three** of the test users were female and **twelve** of the test users were male.
- All users were computer literate.
- Each user had a good level of IR search experience.

Clearly our target user for ostensive and object-based video shot retrieval is somebody who should be a trained professional in the area and we have not targeted the naive casual user. This is deliberate.

- All users were provided with 35 minutes of training on 5 example topics.
    - The system developer did 2 of the example topics.
    - 3 of the example topics were done by the user themselves under the supervision of the system developer.
- Each of the three evaluation systems was used in rotation by the user.
- Each user was given 12 evaluation topics.
- Each user was given 7 minutes to complete each topic.

The query topics were given to the user in three possible query layouts (ascending, descending and split) and three possible system order rotations (first, second or third). The reason for the rotations is to reduce the effect of individual query and user bias in judging system performance.

As people work with an application over time their performance can increase due to the experience they have gained, this could lead to lower performance for early queries than later ones in our experiment. A variation in query layout can reduce this effect by not showing the query topics in the same order for all users and this is a principle of good experimental design which we have incorporated.

Each user evaluates three experimental systems on rotation over all 12 queries and this is done so that a higher performing user would not unduly boost the performance of any one system. In order to ensure that all systems do all query topics, the start point of the rotation of each system is also varied for all users.

Table 6-1 shows each user, the query topic layout and the query topic system order designed for our experiments.

Table 6-1: User Query Topic and System Order

User Number	Query Topic Layout	System Repeat Order
1	Ascending (Topic 1-12)	Sys 3,1 then 2
2	Ascending	Sys 1, 2 then 3
3	Ascending	Sys 2, 3 then 1
4	Descending (Topic 12-1)	Sys 1, 2 then 3
5	Descending	Sys 3,1 then 2
6	Descending	Sys 2, 3 then 1
7	Split (Topic 7-12 then 1 – 6)	Sys 1, 2 then 3
8	Split	Sys 2, 3 then 1
9	Split	Sys 3,1 then 2
10	Ascending	Sys 1, 2 then 3
11	Ascending	Sys 2, 3 then 1
12	Ascending	Sys 3,1 then 2
13	Descending	Sys 1, 2 then 3
14	Descending	Sys 2, 3 then 1
15	Descending	Sys 3,1 then 2

An example of query rotation can be seen in Table 6.2 which shows that user 1 and 12 would use the following systems over the 12 evaluation queries:

Table 6-2: User System Example

Query Topic	System
1	3
2	1
3	2
4	3
5	1
6	2
7	3
8	1
9	2
10	3
11	1
12	2

In their task definition, users were asked to find as many shots which were likely to be relevant to the information topic as they could within the 7 minute time limit. They were allowed and encouraged to use a combination of searching and browsing and were allowed to formulate and submit as many searches and RF judgements as they wanted. This meant that users could explore the results of one formulation of the topic as a query and then re-formulate and re-submit if they wanted.

6.4 Query Topic Classes

A query topic can belong to one of three possible class types: narrow, general and broad depending on the number of relevant results in the evaluation corpus. A narrow topic is defined as topic where a correct answer is contained in less than 0.5 percent of the evaluation corpus. A general topic is defined as a topic where a correct answer is contained in greater than or equal to 0.5 percent of the evaluation corpus and less than 1.5 percent of the corpus. Finally a broad query topic is defined as a topic where a correct answer is in more than 1.5 percent of the corpus. There are 20,529 shots in the evaluation corpus therefore:

- 102.64 shots is equal to 0.5 percent of the corpus
- 205.29 shots is equal to 1 percent of the corpus
- 307.93 shots is equal to 1.5 percent of the corpus

Table 6-3 shows how the three topic classes are calculated:

Table 6-3: Topic Class Calculation

Topic class	The total number of relevant shots in a collection
Narrow	Less than 102.64 Shots
General	Greater than or equal to 102.64 Shots and less than 307.93 Shots
Broad	Greater than or equal to 307.93 Shots

From Table 6-4 we can see the 12 query topics, the number of relevant shots in the corpus, their topic class type and the topic description.

Table 6-4: Evaluation Query Topics

Query Number	# Rel Shots	Class Type	Topic Description
1	149	General	Find shots of any Simpsons character smoking. Shots where the character is holding a cigarette are also relevant.
2	153	General	Find shots of people hurt and in pain. The sound must be audible.
3	43	Narrow	Find shots of Bart, Lisa and Maggie together in the sitting room WITHOUT Homer. Shots can contain other characters but must contain Bart, Lisa and Maggie and NOT Homer
4	221	General	Find shots of Homer working at the nuclear power plant. Valid results can show him at his desk or 'working' around the plant. Shots in the plant canteen are also valid.
5	225	General	Find shots of Bart Simpson sad, upset, or worried. The emotion should be evident on his face and/or in his voice.
6	22	Narrow	Find shots that show Bart and a teacher. Valid results will show JUST Bart and a teacher, no other character should be visible.
7	45	Narrow	Find shots that show an explosion. The sound must be audible.
8	349	**Broad**	Find shots that show Homer's neighbour Ned Flanders.
9	62	Narrow	Find shots of the 'new' owners of the Springfield nuclear power plant when Mr Burns decided to sell. Note: Shots of Mr Burns alone are not valid. Valid results are any of the new owners.
10	155	General	Find shots of the Simpsons character that proposes to Marge's sister Selma. Will a number of characters proposed there is only ONE correct answer in this content.
11	631	**Broad**	Find shots of video that feature any sporting event. A valid result can show a sporting event taking place and/or being watched by any of the Simpson's characters.
12	99	Narrow	Find shots of Homer and Marge in the Kitchen without Bart, Lisa and Maggie.

With a seven minute time limit per topic it is clear that for some topics, such as topic 11 with 631 known relevant shots, not all relevant shots could possibly be found whereas for others such as topic 6 with only 22 relevant shots, locating all shots is a target. In the case of the first type of topic the end-goal is to find as many of the 631 as a user can (since all relevant shots are treated equally) and so higher precision is the target, while for topics like topic 6 a combination of high precision and high recall is the target.

6.5 Overall Retrieval Performance for each System

In this section we will compare the performance of the three systems on all 12 query topics. As we discussed in the previous chapter the first system is a non-ostensive system (sys_non) whereas the second (sys_ost_1) and third (sys_ost_2) are both ostensive systems with a variation in decay curve.

Figure 6-1 shows the total number of shots returned by users over **all** 12 topics. Remember that each topic is executed 5 times per system for each of 3 systems and the results shown are an overall total of shots returned for all query topics by system. As we can see from the results, sys_non has the highest overall number of shots returned at 2447 which is 10.1 percent higher than sys_ost_1. The second system (sys_ost_1) is in turn 4.5 percent higher than sys_ost_2.

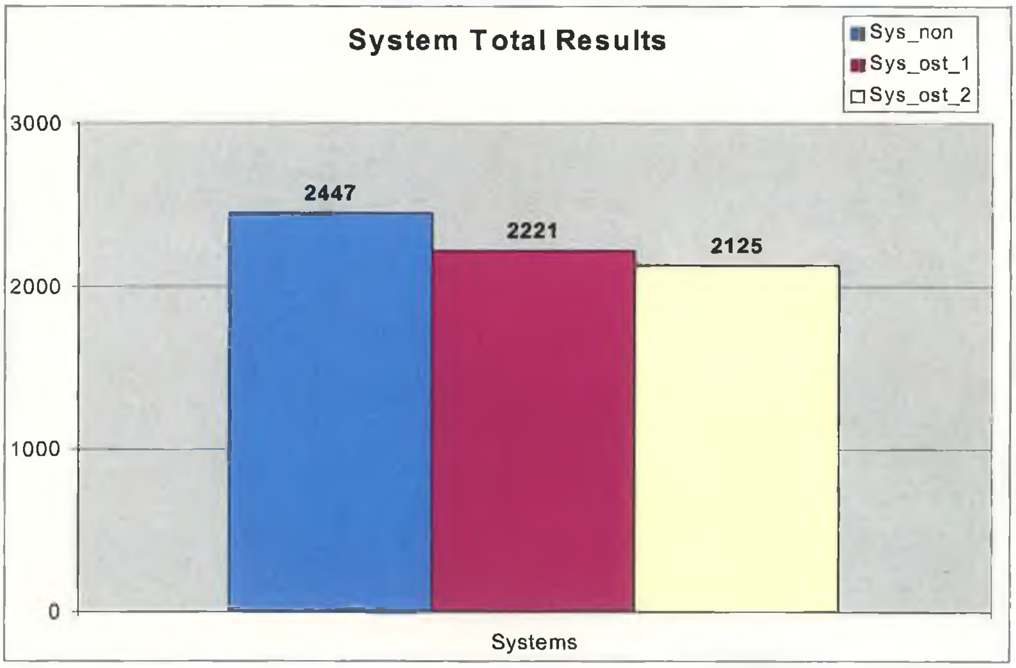


Figure 6-1: Overall Results for each System

Figure 6-2 shows the number of shots returned by users of each system for each of the 12 topics. As we can see for 6 of the 12 topics (1, 2, 4, 5, 8 and 11) there is a large gap between the number of shots that were found by the users and what could have been achieved (shown with an orange bar). The results also show that sys_non does not always have the most results added, for example in topics 1, 3 and 8, where sys_ost_1 returned most results.

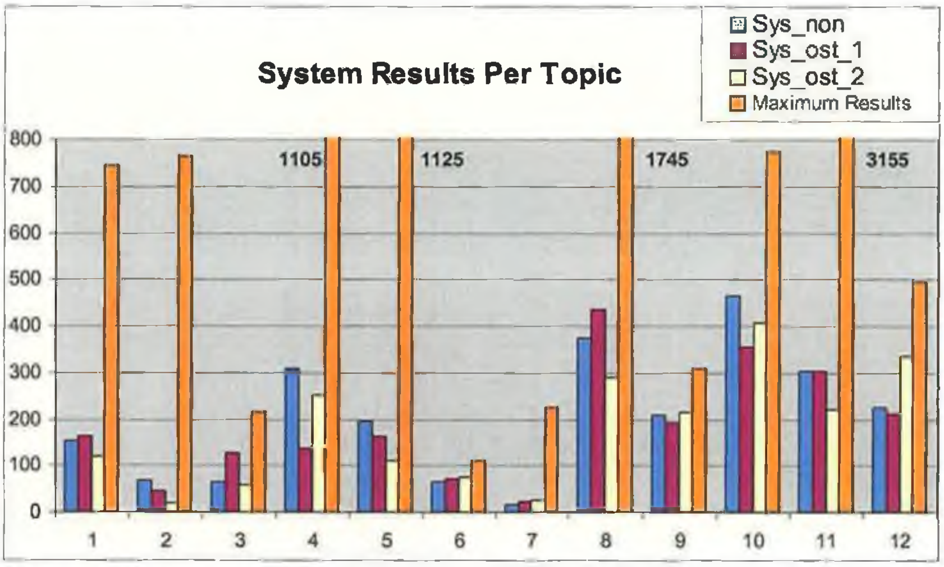


Figure 6-2: System Results Per Topic

The results show that users operating sys_non are finding more results than with the two ostensive systems. It is not clear yet if this means that more relevant results are being found by the first system, just that more shots the users think are relevant, are being found.

Next we look at the **relevance** of the returned results. As we can see from Figure 6-3 the combined results for all topics show that the number of relevant results for sys_non is significantly better (22.3 percent) than the two ostensive systems. The two ostensive systems perform comparably over the 12 query topics differing by only .29 percent. The difference drop between results submitted and their relevance seems to be similar for sys_non and sys_ost_1 whereas sys_ost_2 seems to exhibit less of a drop.

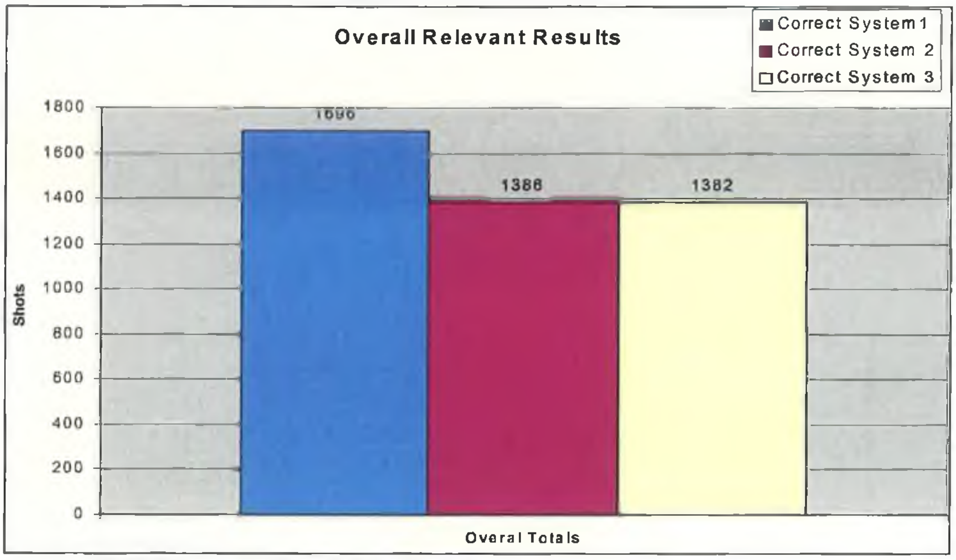


Figure 6-3: Overall Relevant Results

When we look at retrieval performance in terms of overall recall (Figure 6-4) we can see that sys_non is actually 7.2 percent higher than the two ostensive systems overall.

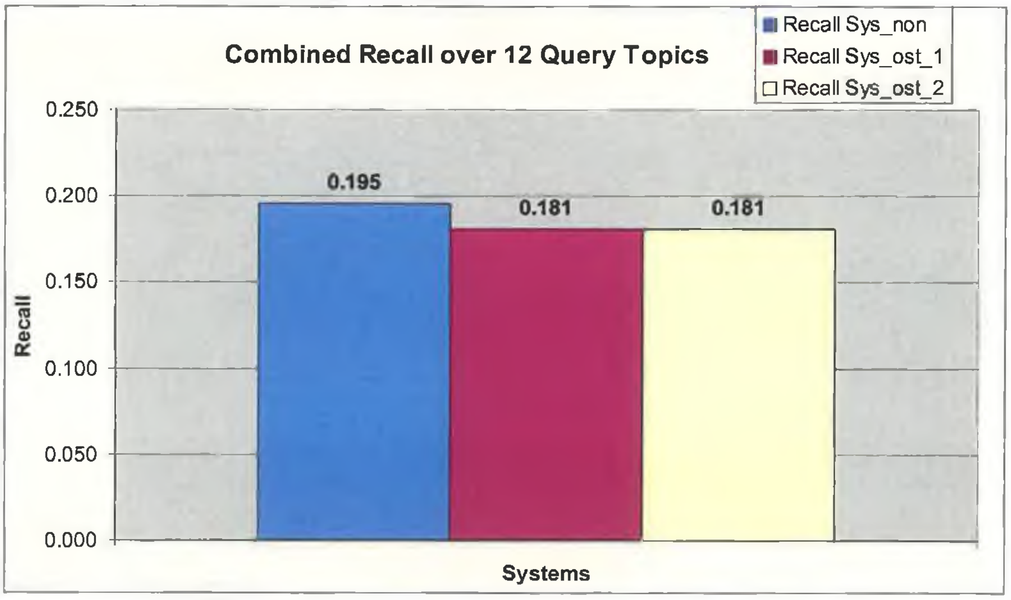


Figure 6.4: Combined Recall over 12 Query Topics

Figure 6-5 shows the overall precision results for each of the three systems and the results indicate that sys_non performs 2.3 percent better than sys_ost_1 and 6.4 percent better than sys_ost_2. There is a very large difference in accuracy between the sys_non and sys_ost_2, and we will investigate this further when we look at individual topic performance. Later we will also examine why the Precision is not higher for all systems, i.e. why users are submitting some results which are not actually relevant.

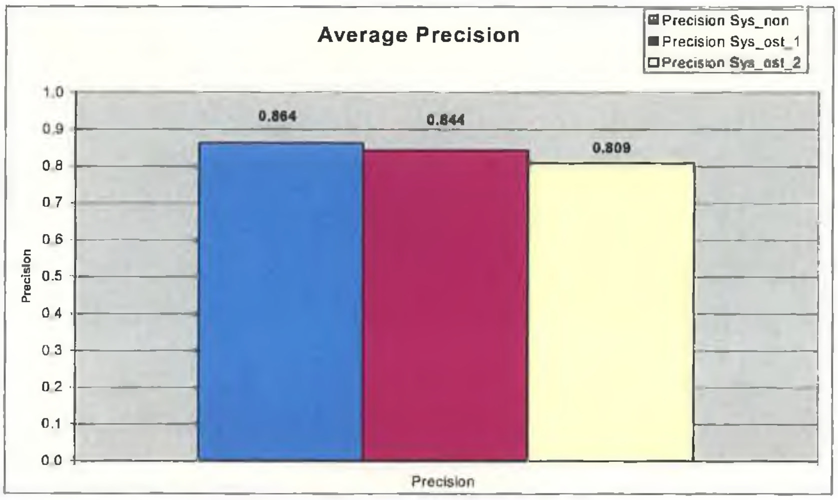


Figure 6-5: Average System Precision

6.6 Topic Retrieval Performance for each System

In this section we will look at system performance on a topic by topic basis. As a result of the experiment's query and system rotations each of the 12 queries is run 5 times by users for each system (12 query topics * 3 systems * 5 users). The results shown are an overall total for each topic and system.

Figure 6-6 shows the recall levels for each topic and system. The results indicate that sys_non offers improved performance on 6 of the 12 topics (1, 2, 4, 5, 8 and 11), sys_ost_1 offers improved performance on three topics (3,7 and 9) and sys_ost_2 offers improved performance on three topics (6,10 and 12). The results indicate that both ostensive systems (sys_ost_1, sys_ost_2) seem to offer improved recall performance on the narrow topics and we will look at this in more detail later.

This result is actually interesting because in Figure 6-4 the aggregate Recall figures for different systems hide the fact that the ostensive-based systems perform better than the non-ostensive one for half the topics. This makes it worthwhile to look at the per-topic Precision performance figures.

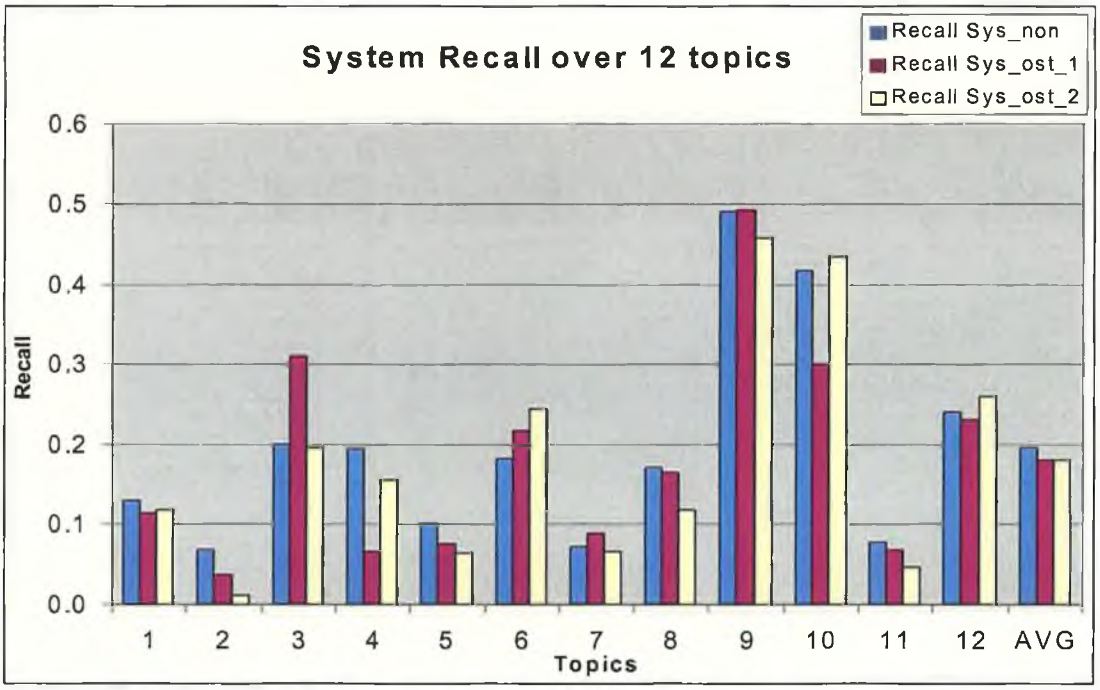


Figure 6-6: System Recall for each Topic

In Figure 6-7 we can see the precision for each system per topic, or in other words the proportion of relevant shots in the set retrieved by users for each topic. The first system (sys_non) offers improved precision for 8 of the 12 topics (2, 3, 4, 7, 9, 10, 11 and 12). The second system (sys_ost_1) has improved precision on 2 topics (1 and 6). The third system (sys_ost_2) has improved precision for 2 topics (5 and 8).

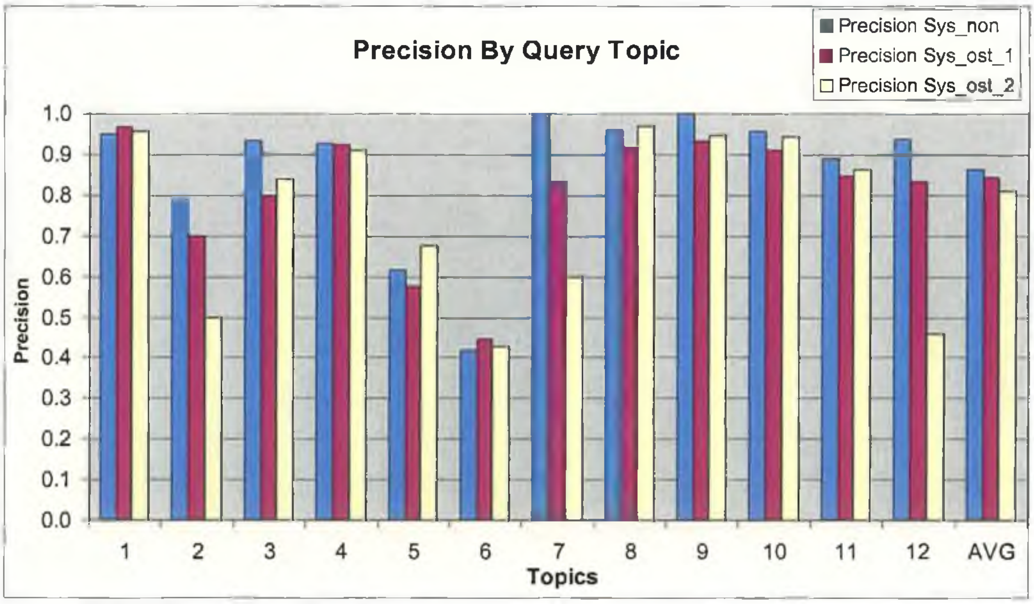
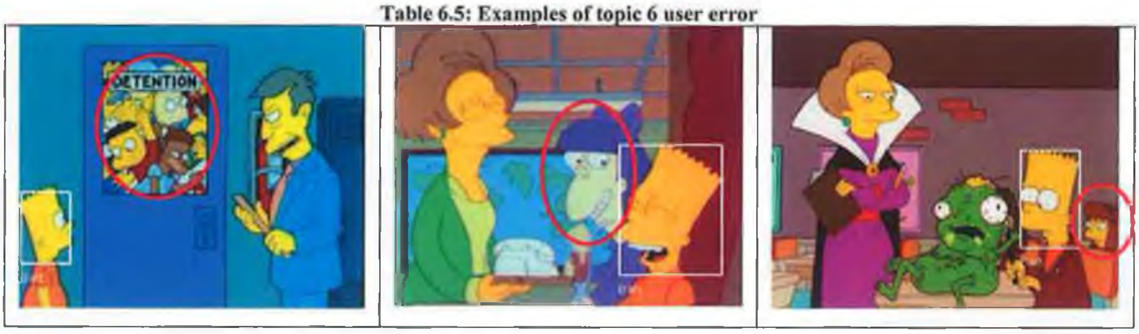


Figure 6-7: System Precision for each Topic

The performance shown in Figure 6-7 is low for topics 2, 5 and 6 with precision of less than 80 percent for all system variations. When we look at the information need in query topic 2 "*Find shots of people hurt and in pain. The sound must be audible*" and topic 5 "*Find shots of Bart Simpson sad, upset, or worried. The emotion should be evident on his face and/or in his voice*" the requirement in both topics is for accurate reading of emotion by the users. The emotion requirement is more ambiguous and challenging than a straightforward topic like "*Find shots that show Homer's neighbour Ned Flanders.*" and this explains why the performance is poor for topics 2 and 5.

Topic 6 "*Find shots that show Bart and a teacher. Valid results will show JUST Bart and a teacher, no other character should be visible*" has a requirement that no other character should be visible, and having studied the output results from users it appears that this requirement has been forgotten in a number of cases. Table 6-5 illustrates three examples of invalid shots that a number of users have included as relevant. The low number of valid results found for the topic assure a low value for precision.

Table 6.5: Examples of topic 6 user error



Sys_ost_2 exhibits very poor precision for the 12th query topic (find shots of Homer and Marge in the kitchen) and this might explain why it has the highest number of false results overall.

Table 6-6 shows the actual number of false results from each of the 5 users who completed the 12th query topic with sys_ost_2. It appears that the non-relevant results for the final user displays an anomaly, as the total of 123 is significantly larger than the previous four users. It would appear that the user had not followed one of the conditions of the query topic correctly perhaps by including results that contained "Bart", "Lisa" and "Maggie" or featured "Homer" and "Marge" in locations other than the kitchen.

Table 6-6: Non-relevant results for the third system on the 12th Query

User	# Non-relevant results	Query Topic 12
1st	18	Find shots of "Homer" and "Marge" in the Kitchen without "Bart", "Lisa" and "Maggie".
2nd	6	
3rd	3	
4th	2	
5th	**123**	

Table 6-7 shows that the large number of non-relevant results returned by the 5th user did not unduly effect the recall performance of sys_ost_2, as the user found a reasonable number of relevant results for the topic. The 5th user obtains a similar relevant shot retrieval performance to the previous four users.

Table 6-7: Relevant results for the third system on the 12th Query

User	# Relevant Results
1st	23
2nd	26
3rd	31
4th	28
5th	**21**

6.7 Narrow, General and Broad Topic Retrieval Performance

In the previous section we have seen that the non-ostensive system appears to perform better than the ostensive-based systems when looking at results averaged over all 12 query topics. However when we look at the results on a topic per topic basis we can see that this is not always the case, in fact for 50 percent of the queries an ostensive system performs better.

In Table 6-3 we gave a listing of the 12 query topics and their respective classes (narrow, general and broad). Relevant results in this section will be shown for each of these classes in order to get a clearer picture of how the three systems perform on different topic classes.

The results shown in Figure 6-8 are the recall results for all **narrow** query topics (3, 6, 7, 9 and 12) found with each of the three systems. Firstly, the results indicate that sys_non performs worst or near worst in each case. The average recall performance indicates that sys_ost_1 gives a 12 percent improvement in recall over sys_non and 9 percent over sys_ost_2.

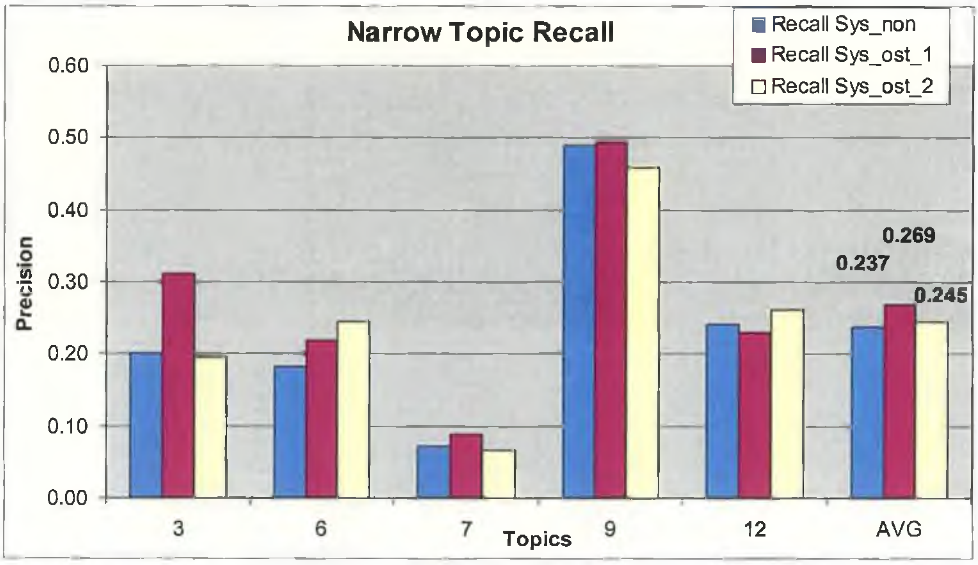


Figure 6-8: Narrow Topic Recall

Precision performance for the three systems on **narrow** topics is shown in Figure 6-9 and demonstrates that for 4 of the 5 narrow topics (3, 7, 9, 12) sys_non offers the highest precision.

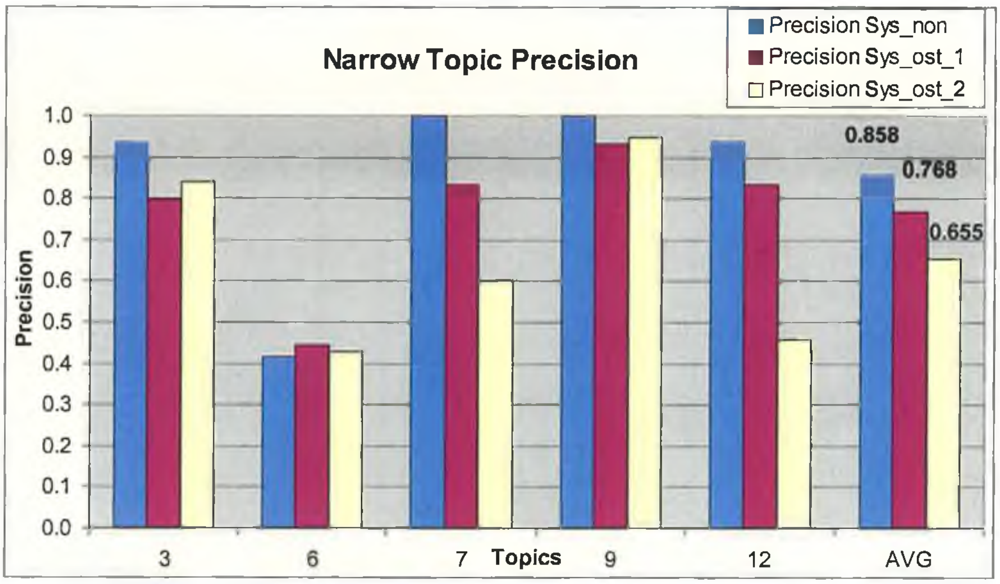


Figure 6-9: Narrow Topic Precision

The results shown in Figure 6-10 illustrate **recall** for all **general** query topics 1, 2, 4, 5 and 10. In nearly every case sys_non offers higher recall than both ostensive-based systems. Query topic 10 is the exception to this rule with a small performance benefit offered by sys_ost_2. Looking at the average recall sys_non offers a 35 percent improvement over sys_ost_1 and 14 percent over sys_ost_2.

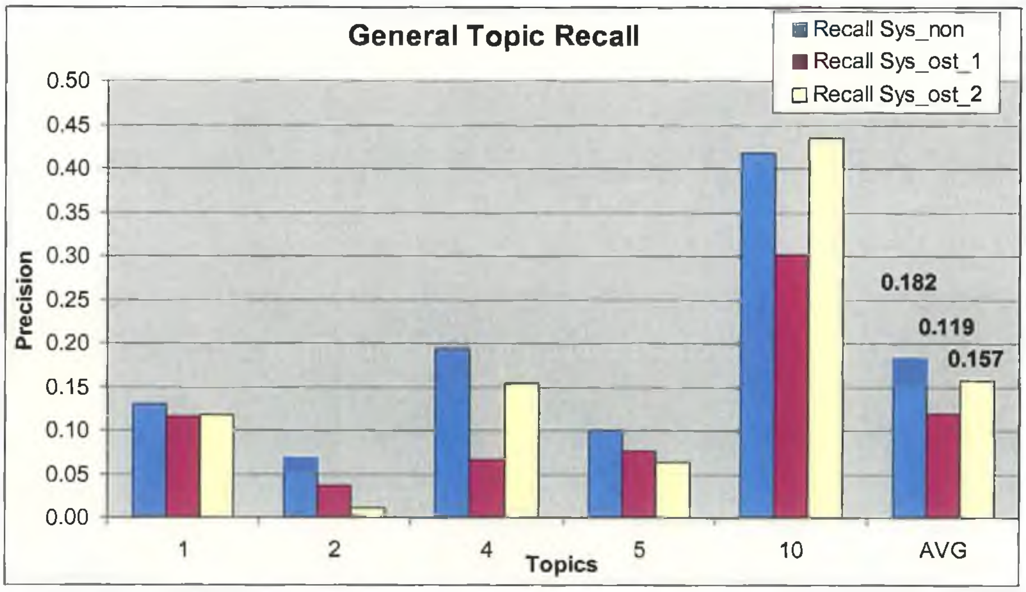


Figure 6-10: General Topic Recall

The **precision** performance for **general** topics is shown in Figure 6-11 and the results demonstrate that sys_non offers 4 percent improvement over system_ost_1 and 6 percent over sys_ost_2.

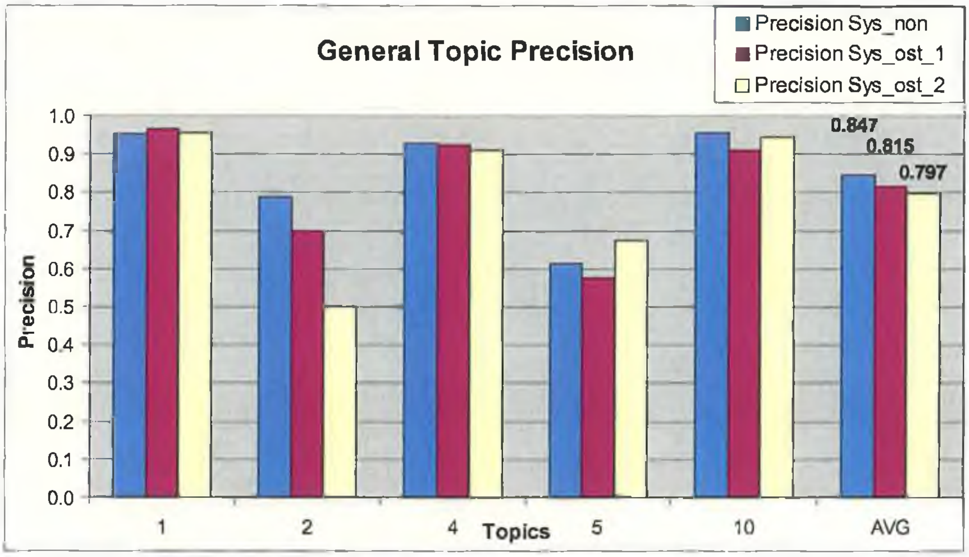


Figure 6-11: General Topic Precision

The results shown in Figure 6-12 illustrate the system **recall** on both **broad** topics. Sys_non again shows improved recall performance over both two ostensive systems. Sys_non offers 7 percent improved recall over sys_ost_1 and 34 percent over sys_ost_2. Sys_ost_1 showed lower recall performance than sys_ost_2 for general topics whereas for broad topics the opposite seems to be the case.

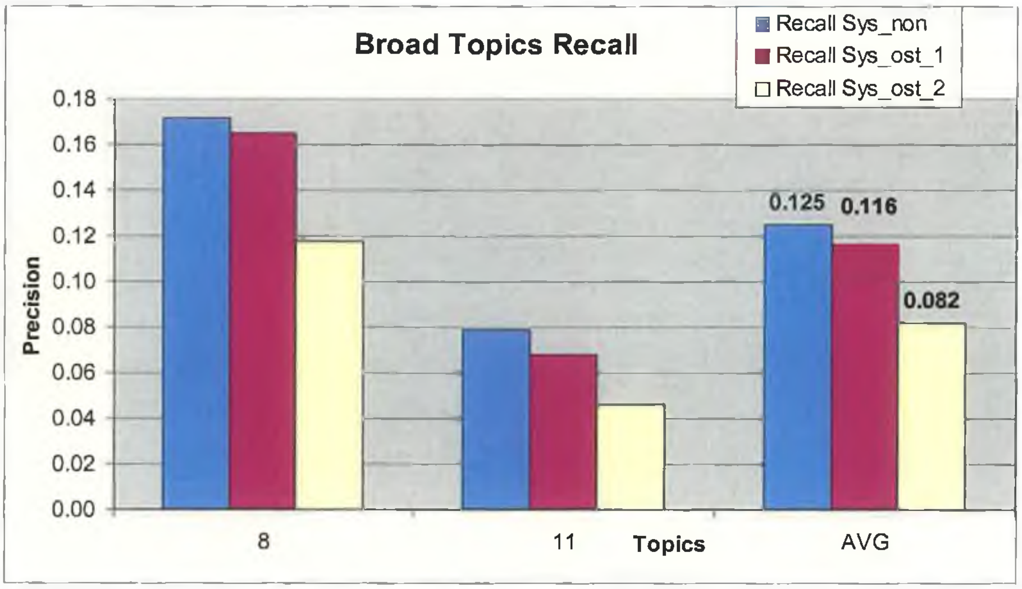


Figure 6-12: Broad Topic Recall

In Figure 6-13 the **broad** query topic **precision** is shown and as we can see from the results performance seems to be similar across all systems. Sys_non offers 4.6 percent improved precision over sys_ost_1 and 0.7 percent over sys_ost_2.

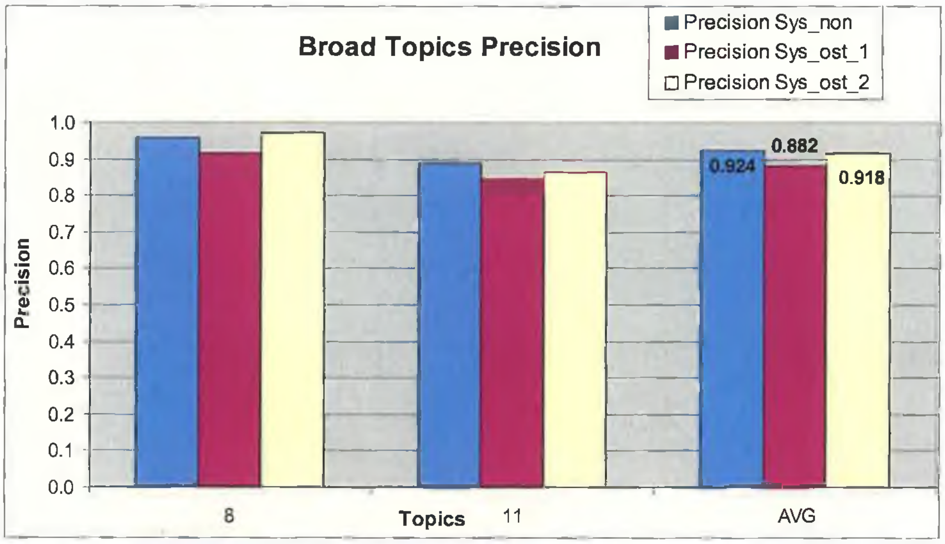


Figure 6-13: Broad Query Topic Precision

6.8 Interactive Experimental Results

The previous section discussed the performance of the three *Físchlár-Simpsons* variants on the 12 query topics. In this section we will discuss the interactive details from the experiment focusing on the following:

- The timing information given by users' experiments, for example the number results added by a user at given minute intervals.
- The usage of two system options: shot context screen and video playback function.
- The usage of four features: object, text, drawing and low-level shot.
- The usage of relevance feedback, for example the average (the maximum and minimum) number of iterations carried out by users.

The results in this section are from the three combined *Físchlár-Simpsons* variants. Individual system variations are not discussed as the systems feature the same GUI.

6.8.1 Timing of Shot Identification

As previously discussed the user is given 7 minutes to answer each query topic. While we know the number of results added by users, the timing of results has yet to be discussed; So, for example, are the results all found in the first three minutes or were most results found at the last minute?

As we can see from Figure 6-14 it appears that a significant number of results were added at each minute of a search topic. The lowest number of results added occurs in the first minute with the highest number of results added by the third minute before declining gradually.

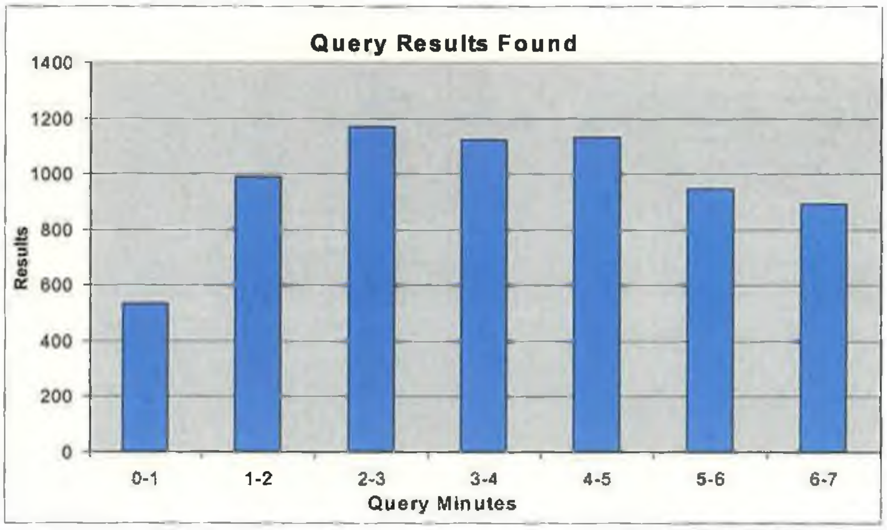
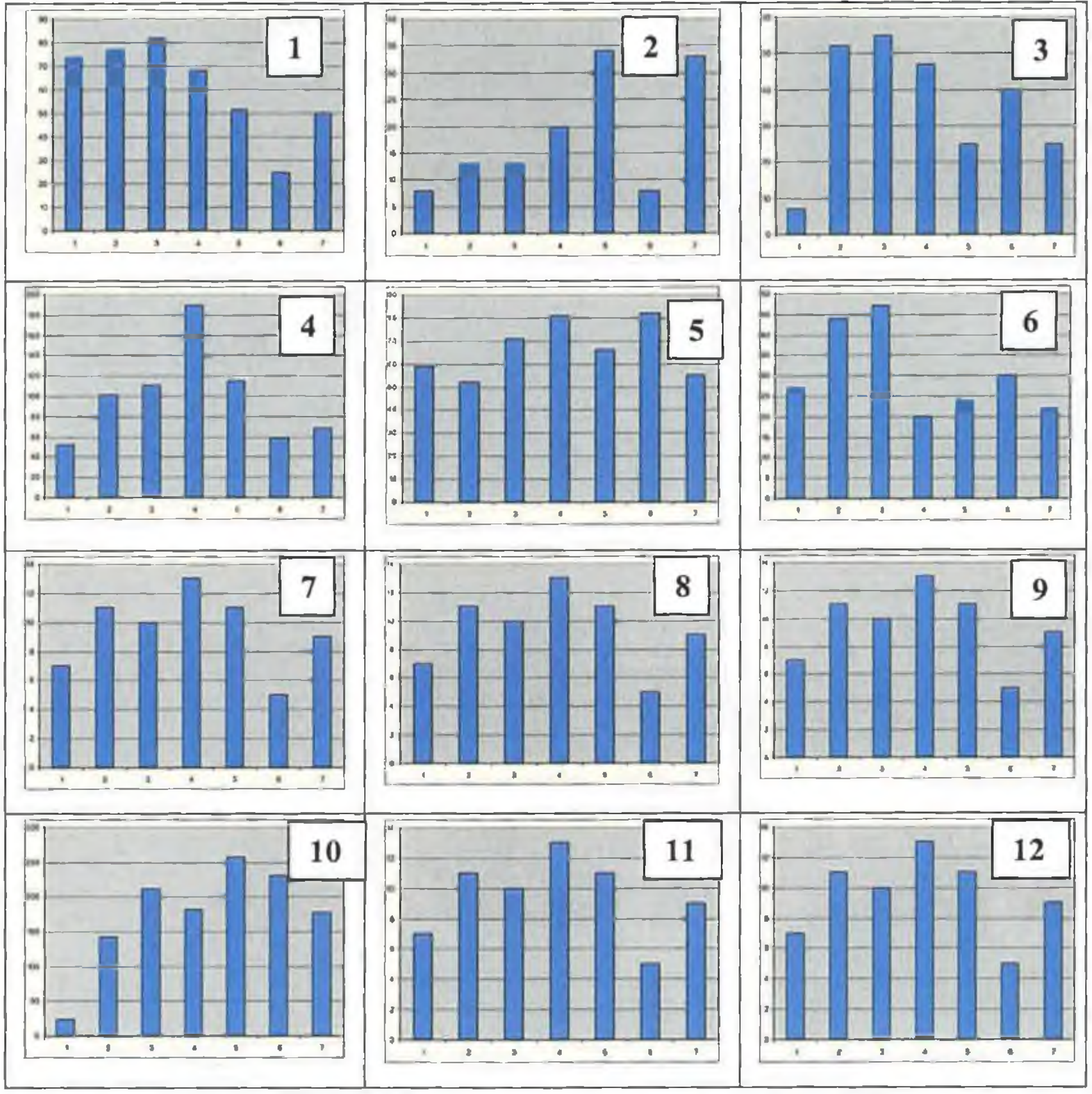


Figure 6-14: Overall results added over the 7 minutes

Table 6-8 shows the timing of user results over the twelve topics and demonstrates that most results are added at the third and fourth minutes followed by a gradual reduction.

Table 6-8: Results Added over 7 minutes for the 12 Topics



The results show that it takes users a minute or two of search before they start to find the bulk of relevant shots, their rate of identification peeks after 3-4 minutes and then starts to tail off.

6.8.2 System Usage

Figure 6-15 shows the number of times the shot **context** screen was selected by the set of 15 users for each of the 12 topics. Topic 2 *"Find shots of people hurt and in pain. The sound must be audible."* and topic 7 *"Find shots that show an explosion. The sound must be audible."* have the lowest use of the context screen. Figure 6-6 showed low recall for both these topics and therefore it is likely that most of the users' time was taken attempting to find valid results.

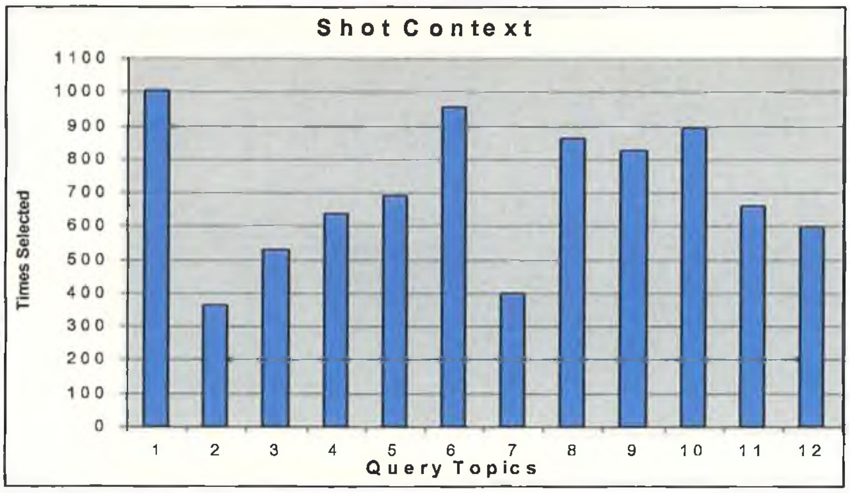


Figure 6-15: Context Screen Usage

Another option available to users was **video playback** and as we can see from Figure 6-16 playback was mainly used on three topics (2, 5 and 7). The reason for the high playback rate for the three topics is that topics 2 and 5 both feature a requirement for emotion in the results and therefore need careful study by the user while topic 7 "*Find shots that show an explosion. The sound must be audible.*" requires the user to carefully study playback to an explosion was audible.

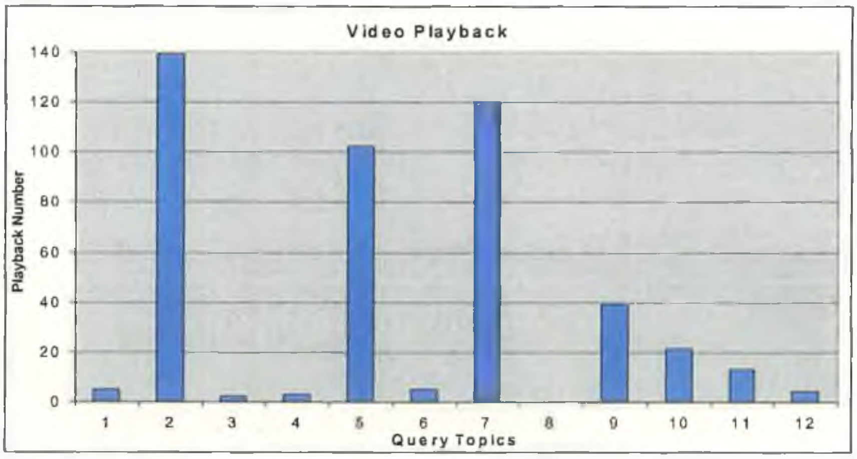


Figure 6-16: Video Playback

6.8.3 Search Feature Usage

In this section we will discuss the usage of the four main search features from *Fischlár-Simpsons* in the experiment in turn: Objects, Text, low-level shots and drawing visualisation. **Note:** 15 users completed 12 query topics each giving 180 as the total number of query topics completed.

Object Search Usage

The general use of objects by users in the experiment is as follows:

• Overall Relevant Objects	: 211	
• Overall non Relevant Objects	: 155	
• Number of queries with objects	: 97	(54%)
• Number of queries with no objects	: 83	(46%)

As we can see objects were used in over 50% of query topics and there was good use of relevant and non-relevant objects as relevance feedback judgements. Figure 6-17 & 6-18 show that object usage was high for topics 3, 4, 5, 6 and 12, and this is to be expected as all five topics mention available objects in the topic description.

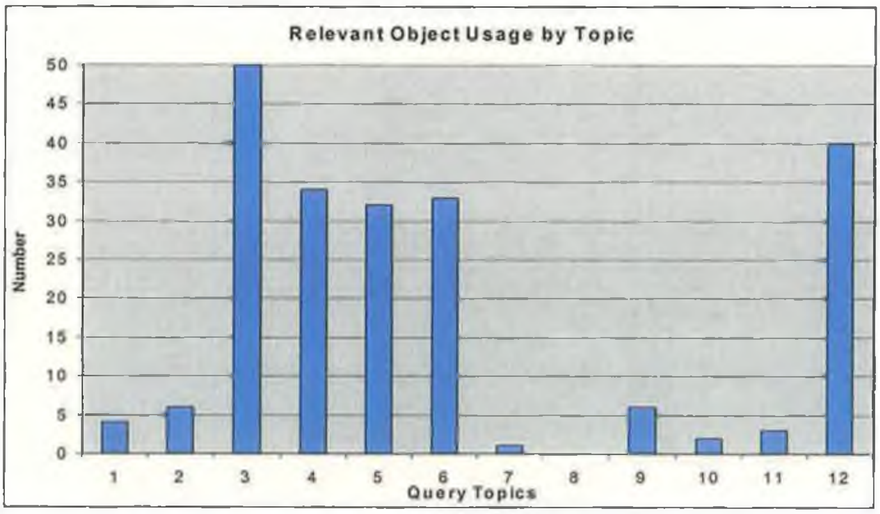


Figure 6-17: Relevant Object Usage by Topic

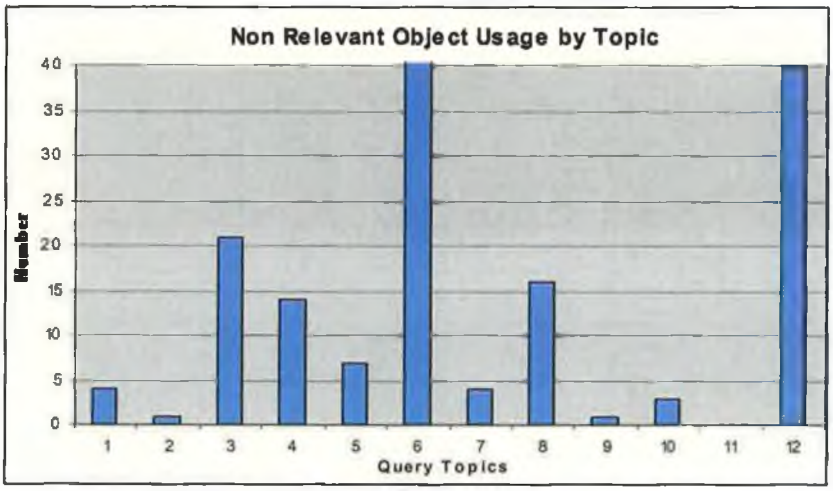


Figure 6-18: Non Relevant Object Usage by Topic

Text Search Usage

The general use of text searching in the experiment is as follows:

• Query topics where text was used	: 175 (97%)
• Query topics where text was not used	: 5
• Overall number of text queries	: 640

Note: 15 Users completed 12 query topics each giving a maximum number of query topics completed of 180. As we can see from the general use of text searching all but 5 user searches featured a text search with an average of 3.5 (640 divided by 180) text searches completed per user topic search.

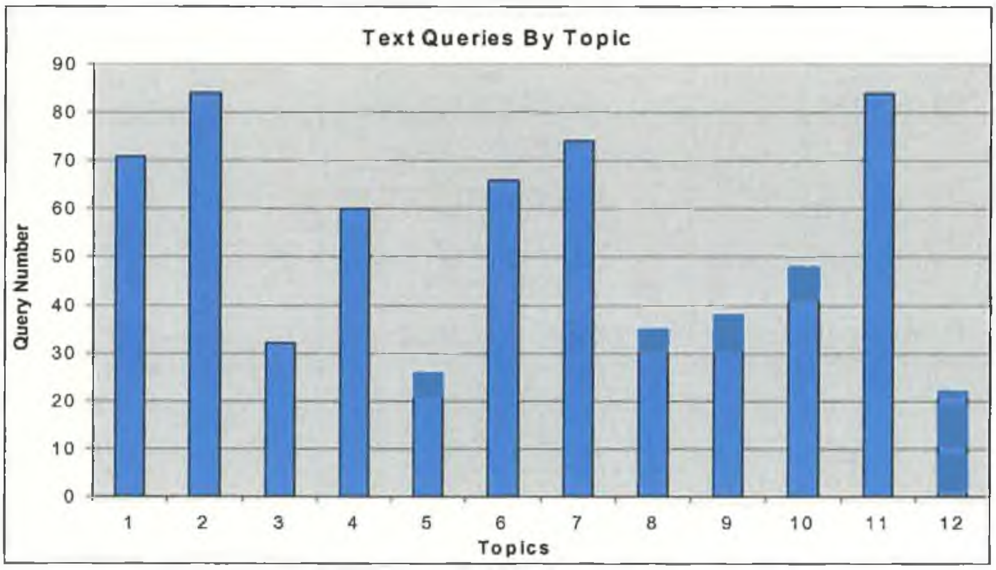


Figure 6-19: Text Queries By Topic

Low-level Shot Similarity Usage

The following is the usage of low-level shot similarity relevance judgements in the experiment:

- Total relevant shot judgements	: 602
- Average judgements per topic (divide by 180)	: 3.3
- Total non relevant shot judgements	: 82
- Total number of queries **with** shot judgements	: 161
- Total number of queries **with no** shot judgements	: 19

Figure 6-20 shows the shot relevant judgements per topic, and we can see that 8 of the 12 topics have over 50 shot judgements for the 15 searches of each topic. Topics 2, 5, 7 and 9 have lower usage of shot judgements due to the nature of the topics. The 2nd and 5th topics have a requirement for emotion where shot similarity will not be an important factor while the 7th topic "*Find shots that show an explosion. The sound must be audible*" is a narrow topic with a low level of relevant results found by users, and therefore it is likely no appropriate RF shot was found by a number of users. Topic 9 is also a narrow topic with all valid shots found in the same general area in the content, so shot similarity would not be of much help here as the valid results would be expected to differ in colour and shape.

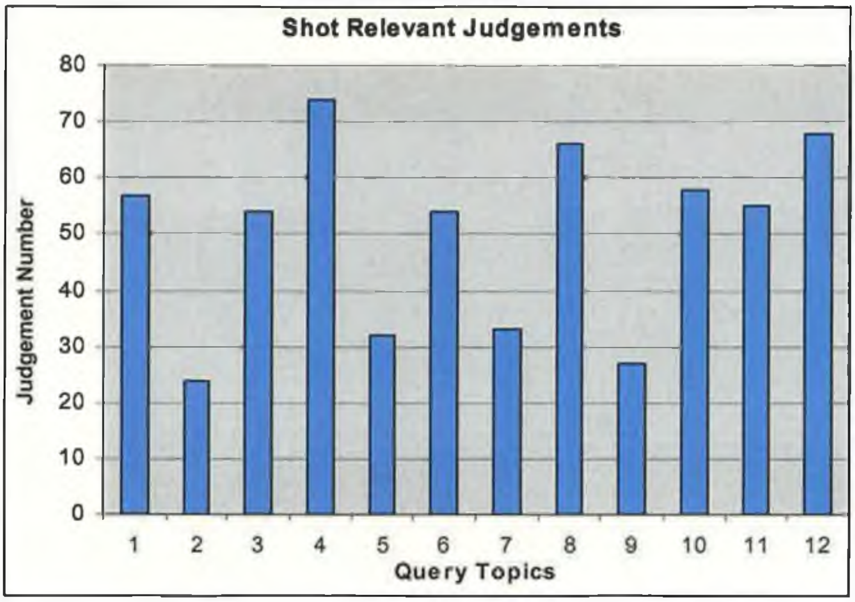


Figure 6-20: Shot Relevant Judgements By Topic

As with object use in searching the number of non relevant shot judgements is much lower than relevant judgements as we can see from Figure 6-21. The two broad topics 8 "*Find shots that show Homer's neighbour Ned Flanders*"and 11 "*Find shots of video that feature any sporting event*" have the highest number of non relevant shot judgements. An efficient search strategy for the two topics would remove shots in the Simpson's family home and other locations by selecting these as non relevant.

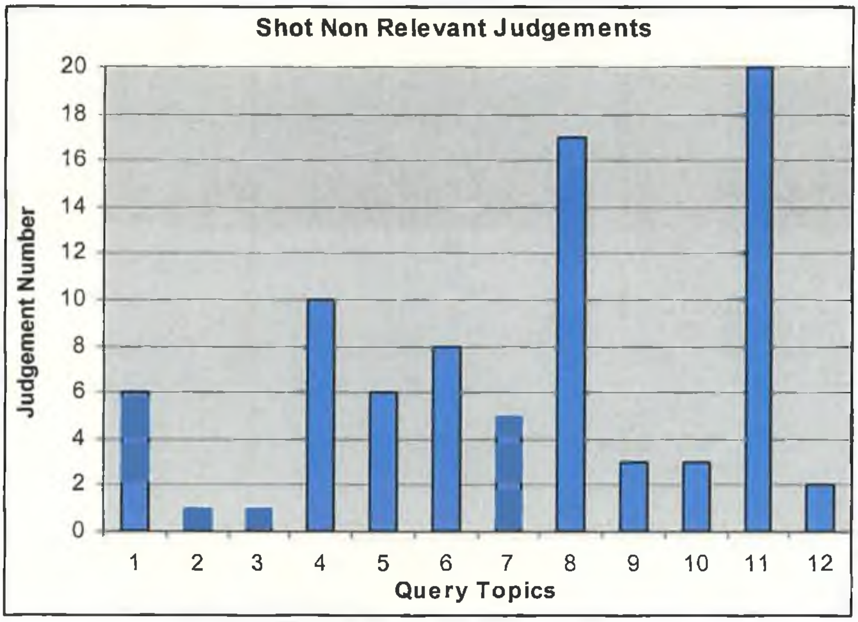


Figure 6-21: Shot Non Relevant Judgements By Topic

Usage of Drawing as a Search Component

The final method of shot ranking is the drawing visualisation approach where the user draws a 'rough' approximation of their information need and this is converted into low level features for shot ranking.

Figure 6-22 shows **low** drawing visualisation usage levels for most topics with topics 7, 8 and 11 the main exceptions. **Topic 7** "*Find shots that show an explosion. The sound must be audible.*" is a narrow topic which has poor level of recall, text search is not effective and relevant examples for shot RF difficult to locate so the drawing offered a final approach that could be employed. **Topic 11** is a search for '*sporting events*' , a search for 'sport', 'football' and 'swimming' finds no relevant results which shows that text search is not very effective for this topic. Text search on **topic 8** *"Find shots that show Homer's neighbour Ned Flanders*" finds numerous valid results when text search is used with the following terms 'Ned' and 'Flanders', perhaps Ned Flander's trademark green jumper (Figure 6-23) might have tempted users to draw the colours with a visualisation approach. Drawing was *not* used for topic 5 "*Find shots of Bart Simpson sad, upset, or worried*" and the availability of a

Bart object search and the difficulty in creating a visual drawing are the main reasons for this. Drawing was not used for topic 9 *"Find shots of the 'new' owners of the Springfield nuclear power plant*" which again is difficult to search using a drawing query.

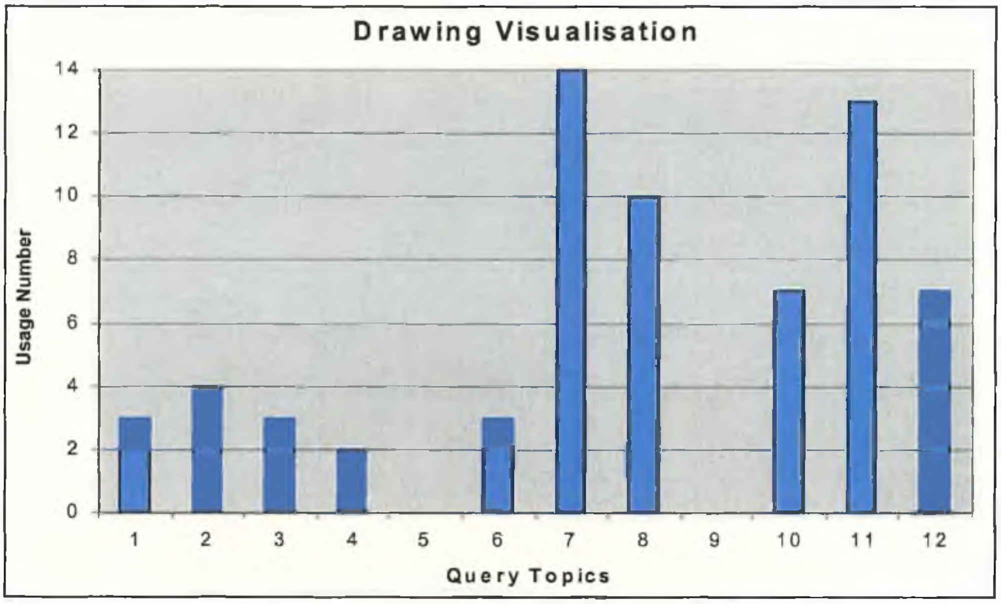


Figure 6-22: Shot Non Relevant Judgements By Topic

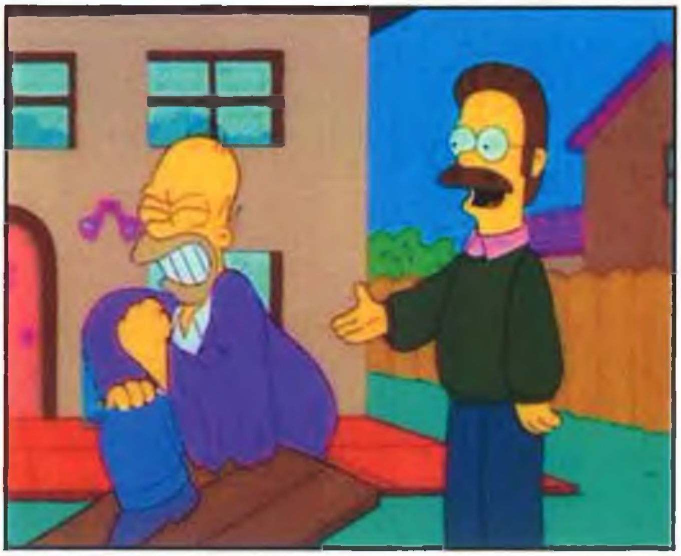


Figure 6-23: Example of a valid result for Topic 8

6.8.4 Relevance Feedback Usage

Relevance Feedback is an important part of *Físchlár-Simpsons* and the effectiveness of the experiment is dependent on the users' RF judgements. The following section discusses users' RF judgements on the three systems over all topics.

Table 6-9 shows the total number of relevance judgements for the four search features at each iteration on the three systems over the 12 topics. Each time a user provides another object, shot, text search or drawing judgement their query expands and the iteration level increments. **Note**: when the user cancels a query and restarts with a new search then the iteration level is reset to zero. Looking at the total judgements we can see that users have tried a number of query search strategies during the 7 minutes as the total judgements is greater than 180 (total number of queries completed by users).

We can see from Table 6-9 that the maximum number of iterations completed by a user is 22. The most popular search strategy employed by users is a text search followed by shot judgements while the second most popular method is a drawing query followed by a text and object search. 80% of user relevance feedback judgements are in the first 8 iterations while 40 % of user relevance feedback is in the first 3 judgements.

Table 6-9: Relevance Judgements by Iteration

RF Iteration	Object	Text	Drawing	Low-level shot	Total Judgements
1	10	***243***	20	1	274
2	58	75	19	***83***	235
3	61	64	2	***75***	202
4	47	56	4	***70***	177
5	39	46	1	***73***	159
6	36	35	5	***65***	141
7	28	25	4	***61***	118
8	18	18	5	***56***	97
9	18	22	0	***41***	81
10	11	13	3	***40***	67
11	14	13	2	***26***	55
12	7	6	0	***30***	43
13	6	7	1	***14***	28
14	3	6	0	***12***	21
15	2	6	0	***9***	17
16	4	0	0	***8***	12
17	2	2	0	6	10
18	1	2	0	6	9
19	0	0	0	5	5
20	1	0	0	2	3
21	0	1	0	0	1
22	0	0	0	1	1

6.9 Summary

The chapter started by discussing three popular IR performance evaluation measures: recall, precision and mean average precision before moving on to discuss the experiment and user details. 15 users were given 12 query topics to complete with 7 minutes allowed per topic and 35 minutes of system training was provided to each user. The three systems under evaluation were given to each user on rotation with topics also rotated in order to reduce individual user and topic bias during the experiment. The 12 query topics are grouped into three classes based on the number of valid results in the corpus: narrow, general and broad.

Taking a look at the overall results the non ostensive RF system obtains improved recall and precision compared to both ostensive systems. When we look at the recall and precision on a topic basis we can see that the two ostensive systems obtain improved recall on 6 of the 12 topics.

In the next section we looked at the precision and recall performance of the three systems on query topic results from the three classes narrow, general and broad. Looking at the average recall on narrow topics, both ostensive systems offer *improved* recall over the non ostensive system while the recall performance of the non ostensive system was *better* than both ostensive systems for general and broad topics.

Having focused on the recall and precision performance of the three systems the next section discussed the interactive usage of the system and in each of the four feature groups (object, text, drawing and low-level shots), differences between systems was not discussed in this section.

The shot context and video playback options were availed of by users during the experiment but their importance tended to be query specific (especially the playback option). In terms of feature usage, objects tended to be topic specific and used in over 50% of query topics. Text search was used in 97% of user query topics with 6 of the 12 query topics featuring a high number of text-based user queries. While low-level shot RF was only used in 89% of user query topics it is the most popular in terms of overall user judgements at 684 compared with 640 for text. The drawing visualisation approach was the least popular approach for most topics, 2 of the 12 topics featured a **large** number of searches using drawing visualisation due to their complexity.

The final section of the chapter discussed the use of relevance feedback by users in the experiment with the results focusing on each of the four search options. The results showed that 80% of user relevance feedback was less than or equal to 8 iterations, 40% of user RF was 3 iterations or less. The most popular search strategy was a text search followed by a number of shot judgements while the second most popular approach was a drawing followed by a text and object search.

In the next chapter we will focus on the main findings from this chapter and discuss possible future research requirements in this area.

Chapter 7 Conclusions and Future Work

7.1 Introduction

In the previous six chapters we described general video IR approaches and a video retrieval model that incorporated variations of relevance feedback and object retrieval. In this chapter we discuss the main issues and findings from each chapter before focusing on the experimental results found in Chapter 6. The final part of the chapter will discuss the main conclusions from this work and possible future research.

7.2 Overall Summary

The start of **Chapter 1** focused on the growth of digital information in the world today and emphasised the need to provide search operations over the content. All digital media types are produced in greater quantities each day with the Internet and consumer requirements fuelling the increase. The storage on standard computers today ranges from 80 to 300 Gigabytes whereas in 1997 that range was 1-2 Gigabytes, with these storage figures it is clear to see why a redesigned file system and digital media search is important. Future computer file systems will need to provide database functionality and performance.

The benefit that text search engines like Google have brought to the Internet is undeniable and it is doubtful that the web would have become so important without them. However, no powerful search tools exist for large quantities of audio-visual media (i.e. video, image and music). While we are able to index and retrieve machine-readable text with a high degree of performance the same is not true for other media types like video, image and audio. The main reason for this is that the semantic information stored in text is explicit while in other media types it is implicit and needs a human to extract the meaning.

Chapter 1 discussed a number of video retrieval systems that facilitate video browsing based on shots and incorporate text search based on associated closed caption text. While the detection of semantic scenes is generally not feasible, the option to segment television news stories (scenes) does exist due to the structured nature of the content.

TRECVID was also described in Chapter 1. TRECVID provides an important common baseline and evaluation for comparing different video retrieval systems. TRECVID has been running since 2001 and each year the complexity of the task and the size of the evaluation corpus has grown. The tasks for 2003 and 2004 focused on ABC and CNN television news content from the late 1990's. A number of query topics were defined each year with example video clips and images representing the information need. TRECVID experiments have shown that quite often text search alone is more effective than image/video search alone and that some topic examples are more suited to visual retrieval than others.

Chapter 2 focused on the various methods of feature extraction that operate on digital video. The main purpose of feature extraction is to provide a searchable index for video content. The extracted features are divided into three main types; visual, audio and temporal. Features can be classified into three different groups based on their semantic levels: low, medium and high.

Text features are considered to be high level and offer the best video retrieval performance. Text-based ASR is human speech from audio which is automatically converted to machine readable text. However, very often important details will be missing from the text, what can be seen on screen, character locations and emotion is not spoken as it is implicitly shown in the visual layer and therefore text searches are not always successful.

Visual features are broken down into colour, edge and object-based. Colour and edge features are considered to be low-level while object-based features are medium-level. The colour histogram is one of the more popular colour features which stores quantities of colour drawn from different colour ranges and uses these in the comparison process. Edges are the boundaries between objects in video and are used in the creation of the edge histogram feature and object detection. The limited quantity of semantic information that we can currently extract from these low level visual features is the main area of difficulty with most offering very poor retrieval performance.

Although object-based features are at a slightly higher level than the colour and edge features, unfortunately as with all object detection approaches only objects that have associated training image examples can be detected which makes it infeasible to detect all objects. Most object detection approaches are concept-based (medium level) detecting specific and concrete objects like cars, planes and trains etc. There are five main reasons given why the detection of objects has proved to be such a difficult challenge: invariance to scale and rotation, occlusion, variation and noise.

Chapter 3 discussed traditional video IR and the main search approaches namely text search, query by example, concept-based and drawing visualisation. Text search is the most popular and effective method of video retrieval due to the discrimination power of the feature. A big disadvantage of a text-based search from ASR is that the text is converted from human audio speech and people's dialogue will very often not mention location, what objects are visible, character names or what event is occurring. Therefore, a text only search can miss a number of relevant results.

Chapter 3 also described the relevance feedback technique for text retrieval. The idea behind this approach is that as documents are ranked, the top N results can be judged relevant or non-relevant by the user. Before re-ranking the results, text features from relevant results are incorporated into current query to form a new query and text features from non-relevant results can be discarded. The process is iterative and as it continues, the query is gradually refined getting closer to the user's information need.

Relevance feedback has been successfully applied to video retrieval systems as the approach can help overcome the poor discrimination power of visual features with the user essentially pointing the system in the right direction based on their relevance judgements. An important text-based RF variation known as ostensive relevance feedback was also described in the chapter. This approach takes into account the order of relevance judgements in a user's search session in the weighting and ranking process. The idea is that as a user starts a query search their information need is vague and somewhat undefined, but as time progresses they are influenced by the ranked results and their query becomes more refined.

Chapter 4 discussed a model of ostensive relevance feedback for video shot retrieval that incorporates text, low-level visual features and template-based object information. Ostensive relevance feedback weights shot judgements based on the order in which they are made and more weight is assigned to newer judgements. Users can include relevant shots and exclude non-relevant shots and choose the relevant features (i.e. visual, object and text) for each iteration. There are two possible ostensive weighting approaches the can be applied: linear and logarithmic based.

Following the introduction of the model, the discussion focused on the approach taken to deal with three main feature types: text, low-level visual shots and template-based objects.

The first step in dealing with closed-caption **text** is to apply stemming and stopword removal techniques to reduce the number of terms before using a **tf-idf** indexing technique to weight the remaining terms. Following this, the remaining terms are aligned at the shot level for retrieval. For text shot comparison a 'value' is given to each term (word) match until a total similarity score is obtained. The final step is to reduce the shot similarity score based on the order of the shot RJ using ostensive weighting.

The **low-level shot** features are applied on a single representative image of each shot (keyframe) to create an index as there would be too much computational expense, redundancy and storage requirements required to index each frame in each shot. For shot comparison the four visual features (colour histogram, edge histogram, regional average colour and median colour) are compared against the comparison shot and generate a dissimilarity score. The final step is to reduce the shot dissimilarity score based on the order of the shot RJ using ostensive weighting.

The template-based **object** feature also indexes the shot keyframe but unlike low-level shot features it generates a similarity score. The index for the template object consists of a listing of keyframes with the name and location of any detectable objects the keyframe contains. A shot similarity score is given based on matching objects present in the shot relevance judgement and the evaluation shot, and this score is increased for a positive judgement and reduced for a matching negative judgement. The facility also exists for similar sized objects and matches found in the same location to also effect the final object comparison score. Having obtained a shot similarity score the score is weighted based on the order of the shot RJ.

The text and object scores are similarity-based whereas the visual score is dissimilarity-based, and what this means in practice is that the complete text and object comparison scores are subtracted from the visual score in order to obtain the final shot comparison score.

Chapter 5 described a video retrieval experiment using *Físchlár-Simpsons* which is an application that incorporates the relevance feedback model described in Chapter 4. The animated television show "The Simpsons" provides the video content used for the experimental evaluation.

Animated content was selected for the experiment as object detection performs better on animated content, and this is due to a number of factors:

- Specific type of colours used. The characters and backgrounds tend to use a repeatable set of colours; in the case of the Simpson's all characters are yellow in colour.
- Well defined character boundaries. The character edges tend to be well defined making the object detection process 'easier'.

- Limited number of detectable poses. The animated characters tend to have a small number of repeated expressions which also makes the detection process more straightforward.

The popularity of the show coupled with the need for animated content on which to run object detection explains why it was chosen for the *Físchlár-Simpsons* experiment. The content for evaluation was divided into two collections: the development corpus which was used for system and feature training and the evaluation corpus that was used in the experiments.

Three *Físchlár-Simpsons* system variations were evaluated in the experiments, two have ostensive RF with a variation in judgement weight decay while the remaining system uses standard RF.

For the experiment **15** users were given training of **35** minutes on the system with 5 training topics before moving on to the actual experiment. The users completed **12** evaluation topics with 7 minutes given to complete each topic. Query topic and system rotation techniques were used to reduce individual user and topic bias in the results.

Chapter 6 discussed three popular IR performance evaluation measures: recall, precision and mean average precision before discussing the results from the users' experiments. The results focus on two main areas: the first is a performance comparison of the three *Físchlár-Simpsons* system variations over the **12** topics, while the second discusses the interactive experimental results and focuses on usage of system options, features and users relevance judgements.

The overall results show that the non-ostensive RF system obtained higher recall and precision than both ostensive systems. When the recall results are viewed for each topic, it showed that the two ostensive systems obtain improved recall on 6 of the 12 topics.

When we looked at the precision and recall performance of the three systems based on three classes of query topics (i.e. narrow, general and broad), it showed a number of interesting differences. The average recall of both ostensive systems on narrow topics showed improved performance over the non-ostensive system while the recall performance of the non-ostensive +system was better than both ostensive systems for general and broad topics.

The analysis of the interactive experimental results also showed a number of interesting findings. The usage data focusing on the timing of user results found that results were added over the 7 minutes with the lowest number of results in the first minute and peaking at the 3rd or 4th minute before gradually declining.

Users completed 180 queries (12 topics * 15 users) during the experiment. Text search was used in 97% of query topics, low-level shots were used in 89% while objects were used in 50%.

The usage of relevance feedback shows that 80% of user RF was 8 iterations or less, 40% of user RF was 4 iterations or less, and the highest number of iterations by a single user was 22. The most popular user RF search strategy was a text search followed by a number of shot relevant judgements while the second most popular approach was a drawing followed by a text and object search.

7.3 Conclusions

The main aim of this work is to probe the benefit of ostensive relevance feedback and of object retrieval, and the *Físchlár-Simpsons* system experiment was developed in order to explore some issues related to this. In this section we will discuss the main findings from the research.

The use of digital video content is growing and improved methods of searching are becoming more important. As the storage costs reduce and capacity increases the need for improved search options will become even more important.

While the feature of text extracted from the audio layer of video offers high retrieval performance, a visual feature at the same level does not yet exist. Visual features tend to offer poor retrieval performance due to their lack of semantic information. Visual features that do contain semantic information tend to be domain restricted or concept-based.

Video relevance feedback can help overcome the poor discrimination power of visual features with the user essentially pointing the system in the right direction based on their relevance judgements.

The experimental results indicate that the non-ostensive system performs higher than both ostensive systems over the 12 query topics. However, the results provide some evidence that both ostensive systems offer an improvement in recall performance for narrow topics. When a topic is narrow and the users information need is not clearly defined ostensive relevance feedback provides a performance benefit. Ostensive relevance feedback is weighting newer judgements higher than older ones and in a corpus of searchable content where the query topic is narrow with few valid results this is clearly of some benefit. Narrow topics and valid results tend to be more specific and exhibit a lower degree of inter-relationship among user's relevance judgements. New user judgements are the result of a larger amount of query topic experience and influence from the results and therefore are of more value than earlier user judgements.

The findings from the relevance feedback approach in the experiment shows that people favour relevant judgements over non-relevant judgements and 88% of user relevance judgement iterations are 10 or less. The results show that there is a limit in the number of relevance judgements a user is prepared to make for a specific query search strategy. The finding that users also tend to prefer making **relevant** judgements is expected as users can state what they require with greater ease than stating what they do not require.

Text search is the most popular method of query initialisation and used in 97% of query topics. The retrieval performance of text quickly gets people to the relevant areas of the content at which point shot relevance judgements can be used. The result reflects the high retrieval performance offered by text.

When text search failed to find relevant results people were willing to try a drawing visualisation. Topics where text search fails like topic 7 "*Find shots that show an explosion. The sound must be audible*" did not retrieve any relevant results using text and it required a drawing approach to initialise the query search.

Object search was performed in over 50% of queries, and this shows that when character objects are available and suitable for the query topic people will use object search.

The usefulness of video playback during search is topic dependent with only 3 topics in the experiment showing a high use of playback. Shot playback requires more time from the user than a simple glance at a keyframe and as users were under a time limit they needed to be as efficient as possible. This finding leads to the conclusion that in most cases, one keyframe is sufficient to allow users to decide on shot relevance.

7.4 Future Work

The experiment found a number of interesting results and it would be useful to investigate whether the conclusions still hold under different conditions, for example a change in corpus content with a larger number of users. Perhaps a retrieval experiment using natural video might provide different conclusions.

As the experimental results have shown, ostensive relevance feedback offers improved performance over traditional RF for narrow query topics. A possible future area of research would be to analyse users judgements and interaction to automatically decide if the information need is narrow or not and adjust the weighting scheme accordingly. For example, a video RF system could be designed to use non-ostensive RF approach by default, and if a user cannot provide any relevant judgments or only a few within 3 or 4 minutes of the search time, the system would consider that the given topic is narrow enough to automatically switch to ostensive RF approach. Further study is needed to decide on how many relevant judgments would be a sufficient condition for the system to turn off the non-ostensive option.

Another interesting area of research would be a learning approach to users' relevance judgements. As users indicate similarities between shots the system could store the associations and then automatically add associated shots for future similar queries. The learning approach could reduce the number of relevance judgement iterations required by users in future searches.

The importance placed on shot keyframes by users in their relevance decisions was discussed at the end of Section 7.3. In the *Físchlár-Simpsons* system ranked results consisted of keyframe images with a resolution of 150 * 110 pixels and further study on the most effective image size might be useful. When we look at the usage of system playback we can see that for most query topics some playback occurred, and it is possible that the image resolution did not provide enough detail for them.

The length of time given for each query topic has been investigated, and in the experiment 7 minutes was given per query topic but it could just as easily be 6 or 8 minutes. In the experiment users returned results over the 7 minutes with a significant number of results found at the last minute. Users took a minute or two before they started to find valid results in large numbers and in most query topics the maximum number of results peaked at the 3rd or 4th minute before gradually declining. It would be interesting to see if this process still holds if a longer time is allocated to each query or does it adapt to the time given for a query.

References

[Adamek et al., 2003] Efficient Contour-based Shape Representation and Matching, Adamek T and O'Connor N. MIR 2003 - 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, Berkeley, CA, 7 November 2003.

[Amir et al., 2003] IBM Research TRECVID-2003 System. A. Amir, W, Hsu, G. Iyengar, C.-Y.Lin, M. Naphade, A. Natsev, C. Neti, H. J. Nock, J. R. Smith, B. L. Tseng, Y. Wu, D. Zhang. In the proceedings of NIST Text Retrieval Conf. (TREC), Gaithersburg, MD, November, 2003.

[Amir et al., 2001] Towards Automatic Real Time Preparation of On-Line Video. Amir A, Ashour G and Srinivasan S. Proceedings of HICSS-34, January 3-6, 2001, Proceedings of the 34th Hawaii International Conference on System Sciences – 2001.

[Barras et al, 2002] Transcribing Audio-Video Archives. Barras, C., Allauzen, A., Lamel, L., and Gauvain, JL. In Proceedings of ICASSP, pages 13-16, Orlando, May 2002.

[Bates, 1989] The Design of browsing and berrypicking techniques for the online search interface. Bates M.J, Online Review, 13:5, pages 407-424, 1992.

[Browne et al., 2000] Evaluating and Combining Digital Video Shot Boundary Detection Algorithms. Browne P, Smeaton A, Murphy N, O'Connor N, Marlow S and Berrut C. *IMVIP 2000* - Irish Machine Vision and Image Processing Conference, Belfast, Northern Ireland, 31 August - 2 September 2000.

[Browne et al., 2001] Dublin City University Video Track Experiments for TREC 2001. Browne P, Gurrin C, Lee H, Mc Donald K, Sav S, Smeaton A and Ye J. *TREC 2001* - Text REtrieval Conference, Gaithersburg, Maryland, 13-16 November 2001.

[Browne, 2001] Segmenting Digital Video. Browne, Paul. M.Sc. Thesis. School of Computing, Dublin City University, 2001.

[Browne et al., 2002] Dublin City University Video Track Experiments for TREC 2002. Browne P, Czirjek C, Gurrin C, Jarina R, Lee H, Marlow S, Mc Donald K, Murphy N, O'Connor N, Smeaton A and Ye J. *TREC 2002 - Text REtrieval Conference, Gaithersburg, Maryland,* 19-22 November 2002.
Available at: http://www.cdvp.dcu.ie/Papers/TREC2002_paper.pdf

[Browne et al., 2003] Dublin City University Video Track Experiments for TREC 2003. Browne P, Czirjek C, Gaughan G, Gurrin C, Jones G, Lee H, Marlow S, McDonald K, Murphy N, O'Connor N, O'Hare N, Smeaton A, and Ye J. TRECVID 2003 - Text Retrieval Conference TRECVID Workshop, Gaithersburg, Maryland, 17-18 November, 2003.

[Browne and Smeaton, 2004] Video Information Retrieval Using Objects and Ostensive Relevance Feedback. Browne P and Smeaton A. SAC 2004 - ACM Symposium on Applied Computing, Nicosia, Cyprus, 14-17 March 2004.

[Buckley and Salton, 1995] Optimization of Relevance Feedback Weights. Buckley C, Salton G. Proceedings of the 18 Annual International ACM SIGIR Conference., Seattle, USA, pages 351-357, 1995.

[Campbell, 2000] Interactive evaluation of the Ostensive Model using a new test collection of images with multiple relevance assessments. Campbell I. Journal of Information Retrieval, pages 85-112, 2000.

[Canny, 1986] A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, November 1986.

[CDVP, 2004] Centre for Digital Video Processing, Dublin City University, Ireland. Homepage available at: http://www.cdvp.dcu.ie

[CHAUVET] The Chauvet-Pont-d'Arc cave system, South-East France, URL: http://www.culture.gouv.fr/culture/arcnat/chauvet/en/index.html

[Cleverdon et al., 1966] Factors Determining the Performance of Indexing Systems. Cleverdon, C.W, Mills, J and Keen, M. Volume 1, Test Results, ASLIB Cranfield Project, 1966.

[Donnelly et al., 2003] Automatically Detecting Camera Motion from MPEG-1 Encoded Video. Donnelly S, Smeaton A, Berrut C, Marlow S, Murphy N and O'Connor N. IMVIP 2000 - Irish Machine Vision and Image Processing Conference, Belfast, Northern Ireland, 31 August - 2 September 2000.

[Eaking, 1999] Content-based image retrieval - can we make it deliver? Eakins, J P. Presented at CIR-99: The Challenge of Image Retrieval: 2nd UK Conference on Image Retrieval, Newcastle upon Tyne, February 1999.

[Earnshaw, 1985] Fundamental Algorithms for Computer Graphics, Springer-Verlag , Rae A Earnshaw editor 1985. Chapter on Visual Perception and Computer Graphics page 1006.

[efg, 2004] Efg's Computer Lab. Available at: http://www.efg2.com/Lab/index.html, (checked May 2004).

[Flask] Transcoding application, available at: http://www.flaskmpeg.net/ (Last visited August 2004)

[Gauvain et al., 2002] The LIMSI Broadcast News Transcription System. Gauvain JL, Lamel L & Adda G. Speech Communication 2002, volume 37 1-2, pages 89-108.

[Google] Google web search engine. Available at: http://www.google.com

[Google Image] Google image search available at: http://www.google.ie/imghp? (Last visited August 2004)

[Gonzalez et al., 1992] Digital Image Processing. Gonzalez R and Woods R. Addison Wesley, 1992, pages 414 - 428.

[Gurrin et al., 2004] Físchlár-Nursing: Using Digital Video Libraries to Teach Processes to Nursing Students. Gurrin C, Browne P, Smeaton A, Lee H, Mc Donald K and MacNeela, P. WBE 2004 - IASTED International Conference on Web-Based Education, Innsbruck, Austria, 16-18 February 2004

[Harman D, 1992] Overview of the First Text Retrieval Conference. Harman, D. In the Proceedings of the First Text REtrieval Conference (TREC-1), Gaithersburg, Maryland, November 4-6, 1992. Available at: http://trec.nist.gov/pubs/trec1/papers/01.txt

[Harman, 1992] Relevance Feedback Revisited. Harman D. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, SIGIR Forum, June 21-24, 1992.

[Harman, 1992 B] The DARPA TIPSTER project. SIGIR Forum, volume 26(2), pages 26--28. ACM, 1992.

[Heesch and Rüger, 2003] Performance boosting with three mouse clicks - Relevance feedback for CBIR. Heesch D and Rüger S. Proceedings of the 25th European Conference on Information Retrieval Research (ECIR, Pisa, Italy, 14-16 April, LNCS 2633, pages 363-376, Springer-Verlag, 2003.

[Huang et al., 1997] Image indexing using color correlograms. Huang J, Kumar SR, Mitra M & Zhu WJ. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pages 762-768, 1997.

[ICC, 2004] International Color Consortium, Working on Color Profiles that are uniform / CIE compliant. Available At: http://www.color.org

[Informedia] Informedia Digital Video Understanding. Carnegie Mellon University. URL: http://www.informedia.cs.cmu.edu/ (Last Visited September 2004)

[Jarina et al., 2002] Rhythm Detection for Speech-Music Discrimination in MPEG Compressed Domain, Jarina R, O'Connor N, Marlow S, Murphy N. DSP 2002 - 14th International Conference on Digital Signal Processing, Santorini, Greece, 1-3 July 2002.

[Lee et al., 2000] User Interface Design for Keyframe-Based Browsing of Digital Video. Lee H, Smeaton A, Murphy N, O'Conner N and Marlow S. *WIAMIS 2001* - Workshop on Image Analysis for Multimedia Interactive Services, Tampere, Finland, 16-17 May 2001.

[Lee et al., 2001] Físchlár on a PDA: Handheld User Interface Design to a Video Indexing, Browsing and Playback System. Lee H, Smeaton A, Murphy N, O'Conner N and Marlow S. *UAHCI 2001* - International Conference on Universal Access in Human-Computer Interaction, New Orleans, Louisiana, 5-10 August 2001.

[Lee et al., 2003] Mobile Access to the Físchlár-News Archive. Gurrin C, Smeaton A, Lee H, Mc Donald K, Murphy N, O'Connor N and Marlow S. Mobile HCI 2003 - 5th International Symposium on Human Computer Interaction with Mobile Devices and Services, Workshop on Mobile and Ubiquitous Information Access, Udine, Italy, 8-11 September 2003.

[Lin et al., 2003] Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets. Lin, CY, Tseng BL and Smith JR. TRECVID 2003 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, *Maryland*, 17-18 November 2003. Available at: http://www-nlpir.nist.gov/projects/tvpubs/papers/ibm.final2.paper.pdf

[Mc Donald et al., 2001] Use of the Físchlár Video Library System. Mc Donald K, Smyth B, Smeaton A, Browne P and Cotter P. UM 2001 - International Conference on User Modeling 2001, Workshop on Personalization in Future TV, Sonthofen, Germany, 13-14 July, 2001.

[Molich and Nielsen, 1990] Improving a human-computer dialogue. Molich, R., and Nielsen, J. Communications of the ACM 33, pages 338-348.

[MPEG] MPEG home page, URL: http://www.chiariglione.org/mpeg/ (Last checked July 2004)

[MPEG-7] MPEG-7 working documentation, URL: http://www.chiariglione.org/mpeg/working_documents.htm#MPEG-7 (Last Checked July 2004)

[Niblack et al., 2000] Web-Based Searching and Browsing of Multimedia Data, W. Niblack, S. Yue, R. Kraft, A. Amir and N. Sundaresan, IEEE International Conference on Multimedia and Expo, New York, USA, July 2000.

[O'Connor et al., 2000] Físchlár: An On-line System for Indexing and Browsing of Broadcast Television Content. O'Connor N, Marlow S, Murphy N, Smeaton A, Browne P, Deasy S, Lee H and Mc Donald K. *ICASSP 2001* - International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, 7-11 May 2001.

[O'Connor et al., 2003] Region and Object Segmentation Algorithms in the QIMERA Segmentation Platform O'Connor N, Sav S, Adamek T, Mezaris V, Kompatsiaris I, Lui TY, Izquierdo E, Bennstrom CF, Casas JR. CBMI 2003 - International Workshop on Content-Based Multimedia Indexing, Rennes, France, 22-24 September 2003.

[O Hare et al., 2004] A Generic News Story Segmentation System and its Evaluation. O'Hare N, Smeaton A, Czirjek C, O'Connor N, and Murphy N. *ICASSP 2004* - IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Quebec, Canada, 17-21 May 2004.

[Onestat] Internet monitoring. Available at: http://www.onestat.com (Last visited August 2004)

[Park et al., 2000] Efficient Use of Local Edge Histogram Descriptor. Dong Kwon Park, Yoon Seok Jeon, Chee Sun Won. ACM Multimedia 2000, October 30, Los Angeles, California.

[Poynton, 2004] Technical Document: Colour Space Information and Conversions, Charles Poynton, Document available at: http://www.poynton.com/PDFs/coloureq.pdf (Last visited April 2004)

[Rocchio, 1971] Relevance Feedback in Information Retrieval. Rocchio J. Prentice-Hall Inc, 1971.

[Robertson and Sparck Jones, 1976] Relevance Weighting of search terms. Robertson S.E, Sparck Jones K. Journal of the American Society for Information Science, 27th May, pages 129-146, 1976.

[Robertson et al., 1992] Okapi at TREC-3. Robertson S, Walker S, Beaulieu M, Gatford M. Text REtrieval Conference. Proceedings Pages 21-30, 1992. http://research.microsoft.com/users/robertson/papers/trec_pdfs/okapi_trec3.pdf

[Rautianen et al., 2002] Temporal Color Correlograms for Video Retrieval. Rautiainen M, Doermann D. ICPR 2002 August 11-15. Québec Canada. Pages 267-270, 2002.

[Rautianen et al., 2003] TRECVID 2003 experiments at MediaTeam Oulu and VTT. Rautiainen M, Penttilä J, Pietarila, Paavo; Noponen, K.; Hosio, M.; Koskela, T.; Mäkelä, Satu-Marja; Peltola, Johannes; Liu, J.; Ojala, T.; Seppänen, T. TRECVID Workshop at Text Retrieval Conference TREC 2003. Gaithersburg, MD, 2003.

[Salton, 1983] Introduction to Modern Information Retrieval. Salton, G. and McGill, M. McGraw-Hill, 1983.

[Salton and Buckley, 1990] Improving Retrieval Performance by Relevance Feedback. Salton G., Buckley C. Journal of the American Society for Information Science, Volume 41(4), pages 288-297, 1990.

[Sadlier et al., 2003] A Combined Audio-Visual Contribution to Event Detection in Field Sports Broadcast Video. Case Study: Gaelic Football Sadlier D, O'Connor N, Marlow S, and Murphy N. ISSPIT'03 - IEEE International Symposium on Signal Processing and Information Technology, Darmstadt, Germany, 14-17 December 2003.

[Smeaton et al., 2004 B] The Físchlár-News-Stories System: Personalised Access to an Archive of TV News. Smeaton A, Gurrin C, Lee H, Mc Donald K, Murphy N, O'Connor N, O'Sullivan D, Smyth B and Wilson D. RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval, Avignon, France, 26-28 April 2004.

[Smeaton et al., 2004] Experiences of Creating Four Video Library Collections with the Físchlár System. Smeaton A, Lee H and Mc Donald K. Journal of Digital Libraries: Special Issue on Digital Libraries as Experienced by the Editors of the Journal, 2004.

[Smith et al., 1996] Querying by color regions using the VisualSEEk content-based visual query system. J. R. Smith and S.-F. Chang. In Intelligent Multimedia Information Retrieval. IJCAI, 1996. Available at: http://citeseer.ist.psu.edu/smith96querying.html

[Tan et al., 2000] Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation, Yap-Peng Tan, Drew D. Saur, Sanjeev R. Kulkarni, Peter J. Ramadge. IEEE transactions on circuits and systems for video technology, vol. 10, no. 1, February 2000.

[TREC 2003] Overview of the TREC 2003 Web Track. Craswell N, and Hawking D, *The Twelfth Text Retrieval Conference,* NIST Special Publication: SP 500-255.

[TRECVID 2002] The TREC-2002 Video Track Report. Smeaton A, Over P, NIST Special Publication: SP 500-251, *The Eleventh Text Retrieval Conference TREC 2002.* Available at: http://trec.nist.gov/pubs/trec11/papers/VIDEO.OVER.pdf

[TRECVID 2003] TRECVID An Introduction, Smeaton A, Kraaij W, Over P. í, Available at: http://www-nlpir.nist.gov/projects/tvpubs/papers/tv3intro.paper.pdf

[TREC IBM, 2002] IBM Research TREC-2002 Video Retrieval System. TREC 2002 - Text REtrieval Conference, Gaithersburg, Maryland, 19-22 November 2002. Available At: *TREC 2002 - Text REtrieval Conference*, Gaithersburg, Maryland, 19-22 November 2002. Available at: http://trec.nist.gov/pubs/trec11/papers/ibm.smith.vid.pdf

[Van Rijsbergen, 1979] Information Retrieval. van Rijsbergen, C. J. 2nd edition, Butterworths, London, 1979.

[Volkmer et al., 2003] The Moving Query Window for Shot Boundary Detection at trec-12. Text REtrieval Conference TRECVID Workshop, Gaithersburg, Maryland, 17-18 November 2003. Available at: http://www-nlpir.nist.gov/projects/tvpubs/papers/rmit.final.paper.pdf

[Wactlar, 2001] Digital Video Libraries, Wactlar, H.D. *Lecture.* DELOS International Summer School of Digital Library Technologies (ISDL'01), Pisa, Italy, July 9-13, 2001.

[Wactlar, 2000] Informedia - Search and Summarization in the Video Medium, Wactlar, H.D. Imagina 2000 Conference Monaco Jan 31st - February 2000.

[Wang et al., 2001] Video Retrieval and Relevance Feedback in the Context of a Post-Integration Model. Wang R, Naphade M, Huang T.S. 2001 Workshop on Multimedia Signal Processing, October 3-5, CANNES–FRANCE, 2001.

[Xu and Croft, 1996] Query Expansion Using Local and Global Document Analysis. Xu, J., Croft, W.B. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. Zurich 1996.

[Ye and Smeaton, 2003] Aggregated Feature Retrieval for MPEG-7 Ye J and Smeaton A. Poster presented at: 14-16 April 2003. LNCS Series 2633. Springer-Verlag, 2003.

[Yu-Fei et al., 2001] MSR-Asia at TREC-10 Video Track: Shot Boundary Detection Task. Text REtrieval Conference TRECVID Workshop, Gaithersburg, Maryland, November 2001. Available at:
http://trec.nist.gov/pubs/trec10/papers/MSR_SBD.pdf

Appendix A

Table A1: Overall Query Results found per minute

Minutes	Results Found
0-1	532
1-2	987
2-3	1167
3-4	1123
4-5	1132
5-6	945
6-7	893

Table A2: Query Results found per topic

Topic	Minutes	Results Found
1	0-1	74
	1-2	77
	2-3	82
	3-4	68
	4-5	52
	5-6	25
	6-7	50
2	0-1	8
	1-2	13
	2-3	13
	3-4	20
	4-5	34
	5-6	8
	6-7	33
3	0-1	7
	1-2	52
	2-3	55
	3-4	47
	4-5	25
	5-6	40
	6-7	25
4	0-1	52
	1-2	101
	2-3	111
	3-4	189
	4-5	115
	5-6	59
	6-7	68
5	0-1	59
	1-2	52

	2-3	71
	3-4	81
	4-5	66
	5-6	82
	6-7	55
6	0-1	27
	1-2	44
	2-3	47
	3-4	20
	4-5	24
	5-6	30
	6-7	22
7	0-1	7
	1-2	11
	2-3	10
	3-4	13
	4-5	11
	5-6	5
	6-7	9
8	0-1	142
	1-2	173
	2-3	132
	3-4	127
	4-5	225
	5-6	165
	6-7	132
9	0-1	72
	1-2	81
	2-3	122
	3-4	110
	4-5	104
	5-6	77
	6-7	48
10	0-1	24
	1-2	142
	2-3	211
	3-4	182
	4-5	257
	5-6	230
	6-7	177
11	0-1	34
	1-2	146
	2-3	143
	3-4	120
	4-5	114
	5-6	97
	6-7	172
12	0-1	26
	1-2	95
	2-3	170
	3-4	146
	4-5	105
	5-6	127
	6-7	102

Table A3: Usage Stats Relevant Objects by Topic

Query Topic	Relevant Usage
1	4
2	6
3	50
4	34
5	32
6	33
7	1
8	0
9	6
10	2
11	3
12	40
Total	211

Table A4: Usage Stats Non Relevant Objects by Topic

Query Topic	Non Relevant Usage
1	4
2	1
3	21
4	14
5	7
6	44
7	4
8	16
9	1
10	3
11	0
12	40

Table A5: Usage Stats Text

Query Topic	Text Usage
1	71
2	84
3	32
4	60
5	26
6	66
7	74
8	35
9	38
10	48
11	84
12	22

Table A6: Usage Stats Drawing

Topic	Drawing Usage
1	3
2	4
3	3
4	2
5	0
6	3
7	14
8	10
9	0
10	7
11	13
12	7
	66

Table A7: Usage Stats Playback

Topic	Playback Usage
1	5
2	139
3	2
4	3
5	102
6	5
7	120
8	0
9	39
10	21
11	13
12	4

Table A8: Usage Stats Context Screen

Query Topic	Context Used
1	1006
2	362
3	531
4	638
5	690
6	955
7	399
8	863
9	824
10	894
11	660
12	598

Table A9: Usage Stats Relevant Shots

Query Topic	Relevant Shot Used
1	57
2	24
3	54
4	74
5	32
6	54
7	33
8	66
9	27
10	58
11	55
12	68

Table A10: Usage Stats Non Relevant Shots

Query Topic	Non Relevant Shots Used
1	6
2	1
3	1
4	10
5	6
6	8
7	5
8	17
9	3
10	3
11	20
12	2

Table A11: Relevance Feedback Usage Stats

Iteration	Object	Text	Drawing	Shot	Total
1	10	243	20	1	274
2	58	75	19	83	235
3	61	64	2	75	202
4	47	56	4	70	177
5	39	46	1	73	159
6	36	35	5	65	141
7	28	25	4	61	118
8	18	18	5	56	97
9	18	22	0	41	81
10	11	13	3	40	67
11	14	13	2	26	55
12	7	6	0	30	43
13	6	7	1	14	28
14	3	6	0	12	21
15	2	6	0	9	17
16	4	0	0	8	12
17	2	2	0	6	10
18	1	2	0	6	9
19	0	0	0	5	5
20	1	0	0	2	3
21	0	1	0	0	1
22	0	0	0	1	1
23	0	0	0	0	0
24	0	0	0	0	0

Table A12: Relevance Feedback Usage Topic 1

Query	Iteration	Object	Text	Drawing	Shot
1	1	0	25	1	0
	2	0	8	0	9
	3	1	4	0	10
	4	1	6	1	5
	5	1	5	0	6
	6	2	3	0	7
	7	0	3	0	6
	8	1	2	0	5
	9	1	5	0	2
	10	1	2	1	3
	11	0	3	0	3
	12	0	1	0	3
	13	0	1	0	1
	14	0	1	0	1
	15	0	2	0	0
	16	0	0	0	1
	17	0	0	0	1
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A13: Relevance Feedback Usage Topic 2

Query	Iteration	Object	Text	Drawing	Shot
2	1	0	27	0	0
	2	1	9	2	9
	3	2	10	1	2
	4	1	9	1	2
	5	0	8	0	3
	6	1	7	0	2
	7	1	5	0	1
	8	0	4	0	3
	9	1	3	0	1
	10	0	2	0	0
	11	0	0	0	1
	12	0	0	0	1
	13	0	0	0	0
	14	0	0	0	0
	15	0	0	0	0
	16	0	0	0	0
	17	0	0	0	0
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A14: Relevance Feedback Usage Topic 3

Query	Iteration	Object	Text	Drawing	Shot
3	1	1	14	1	1
	2	11	2	0	3
	3	13	1	0	2
	4	12	1	0	3
	5	6	4	0	6
	6	8	2	0	6
	7	6	0	0	9
	8	4	1	1	4
	9	4	1	0	5
	10	2	1	1	4
	11	1	2	0	4
	12	1	0	0	5
	13	1	1	0	2
	14	0	1	0	1
	15	1	1	0	0
	16	0	0	0	0
	17	0	0	0	0
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A15: Relevance Feedback Usage Topic 4

Query	Iteration	Object	Text	Drawing	Shot
4	1	6	17	1	0
	2	8	6	1	8
	3	5	6	0	9
	4	6	6	0	7
	5	6	4	0	8
	6	3	5	0	9
	7	2	3	0	9
	8	0	2	0	10
	9	4	1	0	7
	10	0	2	0	7
	11	3	2	0	4
	12	2	1	0	3
	13	1	2	0	0
	14	0	1	0	1
	15	0	1	0	1
	16	1	0	0	1
	17	1	0	0	0
	18	0	1	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A16: Relevance Feedback Usage Topic 5

Query	Iteration	Object	Text	Drawing	Shot
5	1	0	17	0	0
	2	13	1	0	2
	3	6	4	0	3
	4	3	3	0	4
	5	1	1	0	5
	6	2	0	0	3
	7	1	0	0	4
	8	1	0	0	3
	9	2	0	0	2
	10	1	0	0	3
	11	3	0	0	1
	12	1	0	0	2
	13	1	0	0	2
	14	1	0	0	1
	15	0	0	0	2
	16	1	0	0	0
	17	1	0	0	0
	18	1	0	0	0
	19	0	0	0	1
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A17: Relevance Feedback Usage Topic 6

Query	Iteration	Object	Text	Drawing	Shot
6	1	1	24	2	0
	2	14	6	0	5
	3	17	4	0	2
	4	8	6	0	9
	5	9	6	0	5
	6	7	4	0	7
	7	4	5	0	7
	8	4	3	1	6
	9	1	2	0	6
	10	4	0	0	5
	11	3	1	0	3
	12	1	1	0	4
	13	2	1	0	1
	14	0	0	0	1
	15	0	1	0	0
	16	1	0	0	0
	17	0	1	0	0
	18	0	1	0	0
	19	0	0	0	1
	20	1	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A18: Relevance Feedback Usage Topic 7

Query	Iteration	Object	Text	Drawing	Shot
7	1	0	25	6	0
	2	0	10	3	8
	3	0	10	0	7
	4	0	8	1	3
	5	0	6	0	6
	6	1	3	2	5
	7	2	2	1	3
	8	1	1	1	3
	9	1	2	0	2
	10	0	2	0	1
	11	0	2	0	0
	12	0	1	0	0
	13	0	1	0	0
	14	0	1	0	0
	15	0	0	0	0
	16	0	0	0	0
	17	0	0	0	0
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A19: Relevance Feedback Usage Topic 8

Query	Iteration	Object	Text	Drawing	Shot
8	1	0	19	2	0
	2	0	4	3	12
	3	1	4	1	12
	4	3	4	1	10
	5	3	2	0	9
	6	2	0	1	8
	7	2	0	1	7
	8	1	0	0	6
	9	1	2	0	3
	10	0	0	0	5
	11	1	0	0	2
	12	2	0	0	1
	13	0	0	1	1
	14	0	0	0	2
	15	0	0	0	1
	16	0	0	0	1
	17	0	0	0	1
	18	0	0	0	1
	19	0	0	0	1
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A20: Relevance Feedback Usage Topic

Query	Iteration	Object	Text	Drawing	Shot
9	1	0	18	0	0
	2	0	5	0	11
	3	2	3	0	6
	4	3	3	0	2
	5	1	2	0	3
	6	0	2	0	2
	7	0	2	0	2
	8	1	0	0	2
	9	0	1	0	1
	10	0	1	0	1
	11	0	1	0	0
	12	0	0	0	0
	13	0	0	0	0
	14	0	0	0	0
	15	0	0	0	0
	16	0	0	0	0
	17	0	0	0	0
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A21: Relevance Feedback Usage Topic

Query	Iteration	Object	Text	Drawing	Shot
10	1	0	20	0	0
	2	0	7	1	10
	3	0	5	0	11
	4	1	5	0	8
	5	0	4	1	9
	6	0	4	2	6
	7	1	1	2	3
	8	1	1	0	2
	9	1	1	0	2
	10	1	0	0	3
	11	0	0	1	2
	12	0	0	0	2
	13	0	0	0	1
	14	0	0	0	1
	15	0	0	0	1
	16	0	0	0	0
	17	0	0	0	0
	18	0	0	0	0
	19	0	0	0	0
	20	0	0	0	0
	21	0	0	0	0
	22	0	0	0	0

Table A22: Relevance Feedback Usage Topic

Query	Iteration	Object	Text	Drawing	Shot
11	1	0	23	4	0
	2	0	13	7	4
	3	2	10	0	8
	4	0	5	0	9
	5	1	3	0	8
	6	0	5	0	5
	7	0	4	0	4
	8	0	4	0	4
	9	0	4	0	2
	10	0	3	1	2
	11	0	2	1	3
	12	0	2	0	4
	13	0	1	0	4
	14	0	2	0	3
	15	0	1	0	3
	16	0	0	0	4
	17	0	1	0	2
	18	0	0	0	3
	19	0	0	0	1
	20	0	0	0	1
	21	0	1	0	0
	22	0	0	0	1

Table A23: Relevance Feedback Usage Topic

Query	Iteration	Object	Text	Drawing	Shot
12	1	2	14	3	0
	2	11	4	2	2
	3	12	3	0	3
	4	9	0	0	8
	5	11	1	0	5
	6	10	0	0	5
	7	9	0	0	6
	8	4	0	2	8
	9	2	0	0	8
	10	2	0	0	6
	11	3	0	0	3
	12	0	0	0	5
	13	1	0	0	2
	14	2	0	0	1
	15	1	0	0	1
	16	1	0	0	1
	17	0	0	0	2
	18	0	0	0	2
	19	0	0	0	1
	20	0	0	0	1
	21	0	0	0	0
	22	0	0	0	0

Appendix B

Simpson's Retrieval: Query Topic Comments & Feedback

Query Topic Number: ____

Did you know the answer to the topic before searching: ____________________

A note on scores (1-10) of **1** indicates **very poor** while **10** indicates **very good**.

Did you use **text** information during search: **Y / N**

If you answered [**Yes**]

How useful was **text** information in this query topic? [ ] (1-10)

Did you use **object** information during search: **Y / N**

If you answered [**Yes**]

How useful was **object** information in this query topic? [ ] (1-10)

Did you use the **drawing query** function during search: **Y / N**

If you answered [**Yes**]

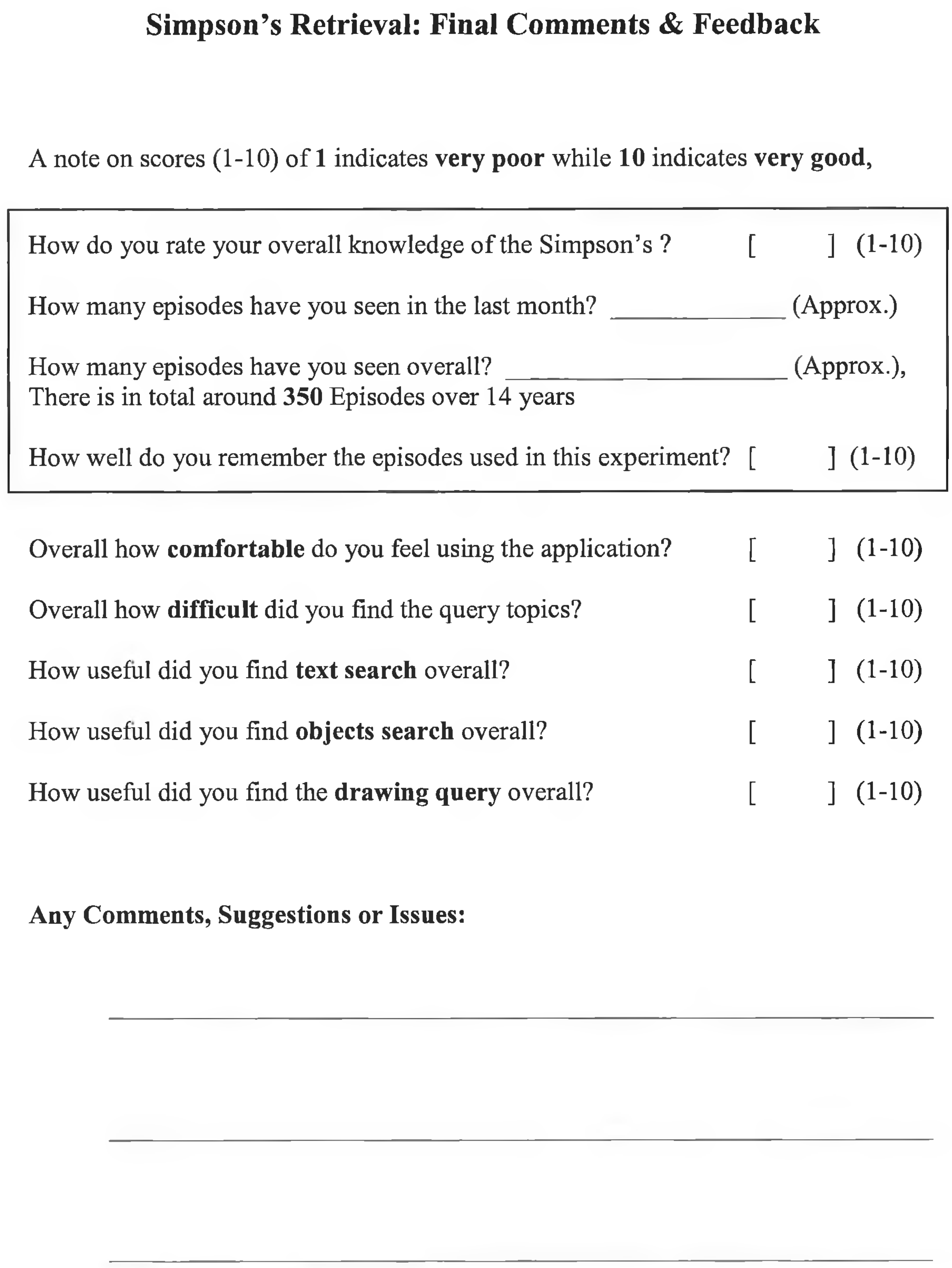
How useful was **drawing function** in this query topic? [ ] (1-10)

How **difficult** did you find answering this query topic? [ ] (1-10)

How **comfortable** do you feel using the application? [ ] (1-10)

Comments: ____

Figure B-1: Simpson's Experiment Topic Questionnaire



Simpson's Retrieval: Final Comments & Feedback

A note on scores (1-10) of **1** indicates **very poor** while **10** indicates **very good**,

How do you rate your overall knowledge of the Simpson's ? [ ] (1-10)

How many episodes have you seen in the last month? ____________ (Approx.)

How many episodes have you seen overall? ____________________ (Approx.),
There is in total around **350** Episodes over 14 years

How well do you remember the episodes used in this experiment? [ ] (1-10)

Overall how **comfortable** do you feel using the application? [ ] (1-10)

Overall how **difficult** did you find the query topics? [ ] (1-10)

How useful did you find **text search** overall? [ ] (1-10)

How useful did you find **objects search** overall? [ ] (1-10)

How useful did you find the **drawing query** overall? [ ] (1-10)

Any Comments, Suggestions or Issues:

____

____

____

Figure B-2: Simpson's Experiment Final Questionnaire

Table B-1: Final Questionnaire User Feedback

User	Knowledge of Simpsons	Episodes seen in last month	episodes seen overall	Knowledge of episodes used in experiments	Comfortable with the system	difficulty	text search	object search	drawing search
1	8	5	200	6	8	6	9	8	4
2	7	15	200	8	8	8	7	8	5
3	5	1	25	1	8	7	8	3	1
4	3	0	3	0	7	5	8	9	4
5	9	4	300	9	10	5	9	9	4
6	8	10	100	6	8	7	7	6	1
7	7	5	170	6	9	6	7	8	5
8	6	12	300	6	6	7	5	8	3
9	8	20	250	7	9	5	7	8	1
10	7	10	200	6	8	7	5	9	4
11	9	20	280	6	10	7	6	8	4
12	0	0	0	2	7	5	8	9	6
13	9	15	250	9	8	7	7	8	4
14	7	12	250	6	7	3	8	6	0
15	9	15	250	7	10	7	9	8	4
16	3	1	15	2	8	9	9	10	6
AVG	6.6	9	175	5.4	8.1	6.3	7.4	7.8	3.5

Table B-2: Final Questionnaire User Comments

User	comment
1	the drawing query was not useful to me for theses particular queries. But I can imagine that it would be useful for other types of searches. Also it was not clear how the search criteria were weighted (e.g. text vs. colour etc). Also in text search, if the word is not found, it should indicate this.
2	I would generally use a relevant key frame as relevant rather than draw my own image. Quite easy to use and fast narrowing the search space.
3	the ranking of shots is changed as soon relevant feedback is given. That makes the user lose the current location in the browsing. There may be more relevant shots at that location. Re-rankings should be used only when users suggest that.
4	
5	a lot of results were as a result of finding a relevant shot and viewing its context to get other shots in the same scene. Once I got used to the system it was easy to use. Generally I found relevant shots quite easily although I generally knew beforehand what i was looking for and where to find it.
6	
7	I was able to find results for most queries. Text seemed good for initial results and the features helped refine search. On some queries, the system responded slowly which was slightly frustrating. Finding objects for object retrieval was sometimes difficult due to the object detection accuracy.
8	have more characters in the object search
9	I thought combination of text and objects was difficult to get right as I found text was given a small weight. Otherwise application was extremely useful in information retrieval, especially object positive and negative feedback
10	the commands (mouse +_ Key board) being non-standard need a bit of getting used to, but were easy once I had. The way you typed in text queries disappears before you can click the query button if your mouse slipped back on the text box, very irritating. context screen most useful. events obviously much harder to find than people based queries. dray and drop would be nice to move shots but then might be more confusing.
11	when a new object or image is selected as negative or positive I found that the browser jumped back to the beginning of the list of images so I found myself having to go through images that I had already browsed. This would not occur if once an image has been selected as a query answer. it is removed from the displayed images.
12	
13	allow multiple selection of objects / frames before updating results. Maybe take into account the certainty of an object being correct, as it could reduce the false-positive objects appearing high in the list of results.
14	
15	I found the most useful part of the application was the context menu - it being available is what made the text search so useful. As the results from the search would serve as a starting point.
16	Objects worked well. I always wanted to use this to reduce non relevant ones. Some topics ask what system can not do, i.e. find a particular audio, is not possible with the system except by browsing.

Table B-3: Topic Feedback User 1

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
1	6	N	7	0	0	7	7
1	7	N	6	3	0	8	6
1	8	N	0	8	6	5	8
1	9	N	8	7	0	6	7
1	10	N	7	8	0	5	7
1	11	N	8	8	0	6	7
1	12	N	8	0	0	5	8
1	13	N	9	0	4	4	8
1	14	y	9	0	0	4	8
1	15	y	9	0	4	5	8
1	16	N	9	0	0	4	8
1	17	N	3	9	0	3	8

Table B-4: Topic Feedback User 2

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
2	7	y	5	0	0	5	7
2	8	y	8	7	0	3	8
2	9	y	8	8	0	2	8
2	10	y	7	0	0	4	8
2	11	y	7	6	0	0	0
2	12	no	5	0	5	5	5
2	13	yes	8	0	0	1	9
2	14	yes	8	0	0	2	9
2	15	no	8	0	0	1	8
2	16	y	7	0	0	3	8
2	17	yes	7	8	0	2	8

Table B-5: Topic Feedback User 3

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
3	7	n	5	1	1	9	7
3	8	y	7	1	1	4	8
3	9	y	9	8	1	2	9
3	10	y	8	7	1	3	9
3	11	n	9	8	1	9	9
3	12	n	7	1	1	9	9
3	13	n	8	1	1	2	9
3	14	n	9	1	1	8	9
3	15	n	9	1	1	1	9
3	16	y	9	1	1	2	9
3	17	y	9	1	1	2	9

Table B-6: Topic Feedback User 4

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
4	7	n	0	0	0	10	7
4	8	n	5	10	0	4	7
4	9	n	5	8	0	4	7
4	10	n	1	10	0	4	7
4	11	n	5	10	0	3	7
4	12	n	2	3	4	7	7
4	13	n	9	8	4	2	7
4	14	n	7	0	0	2	7
4	15	n	5	5	0	6	7
4	16	n	6	0	3	5	5
4	17	n	5	10	0	5	5

Table B-7: Topic Feedback User 5

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
5	7	n	9	0	0	5	10
5	8	y	0	10	0	3	10
5	9	y	6	9	8	5	9
5	10	n	9	9	0	5	9
5	11	y	7	8	0	4	9
5	12	n	2	0	9	7	9
5	13	y	9	0	0	1	9
5	14	y	10	6	0	3	9
5	15	y	9	0	4	7	9
5	16	n	6	8	7	6	9
5	17	y	9	10	0	2	9

Table B-8: Topic Feedback User 6

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
6	7	yes	7	0	0	6	8
6	8	n	6	7	0	2	8
6	9	n	7	7	0	2	8
6	10	n	7	8	0	3	8
6	11	n	6	7	0	7	8
6	12	n	7	0	0	8	8
6	13	n	7	8	0	7	8
6	14	n	5	5	0	7	8
6	15	n	9	0	0	2	9
6	16	n	2	0	8	6	9
6	17	n	6	10	0	1	9

Table B-9: Topic Feedback User 7

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
7	7	n	6	0	0	8	9
7	8	n	6	8	0	7	9
7	9	n	7	8	0	6	9
7	10	n	7	8	0	7	9
7	11	n	6	8	0	7	9
7	12	n	3	4	0	9	7
7	13	n	9	0	0	2	8
7	14	n	6	6	0	7	8
7	15	n	7	0	0	7	9
7	16	n	2	0	7	7	9
7	17	n	7	8	0	7	9

Table B-10: Topic Feedback User 8

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
8	7	n	3	0	2	9	4
8	8	n	4	3	0	7	6
8	9	n	7	6	0	5	7
8	10	n	6	8	0	5	8
8	11	n	2	2	0	9	5
8	12	n	3	0	0	8	7
8	13	n	4	0	6	7	7
8	14	n	7	0	0	8	6
8	15	n	8	0	3	7	6
8	16	n	3	0	8	8	6
8	17	n	4	9	0	5	9

Table B-11: Topic Feedback User 9

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
9	7	n	10	2	0	5	9
9	8	n	1	10	0	5	9
9	9	n	7	8	0	4	9
9	10	n	5	5	0	6	9
9	11	y	3	9	0	4	10
9	12	n	10	0	0	8	5
9	13	y	9	6	0	3	6
9	14	y	10	4	0	4	8
9	15	y	9	7	0	3	8
9	16	n	6	0	0	7	9
9	17	n	3	10	0	8	9

Table B-12: Topic Feedback User 10

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
10	7	n	7	2	0	3	6
10	8	y	3	9	4	6	8
10	9	n	4	9	0	7	9
10	10	n	4	6	0	5	9
10	11	y	4	9	0	4	9
10	12	n	2	0	6	2	9
10	13	y	8	7	0	9	9
10	14	n	5	3	0	1	9
10	15	y	8	0	0	6	9
10	16	n	5	0	8	6	10
10	17	y	4	9	0	9	10

Table B-13: Topic Feedback User 11

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
11	7	n	2	8	0	2	8
11	8	n	2	9	0	6	9
11	9	y	3	8	0	7	9
11	10	n	5	5	0	5	10
11	11	n	7	10	0	5	10
11	12	n	7	2	1	4	10
11	13	y	8	0	0	5	10
11	14	y	6	0	0	0	0
11	15	n	7	0	0	6	10
11	16	n	7	0	0	4	10
11	17	n	0	9	0	8	10

Table B-14: Topic Feedback User 12

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
12	7	n	5	0	0	7	6
12	8	n	8	7	0	5	7
12	9	n	4	6	0	7	7
12	10	n	3	7	0	4	7
12	11	n	5	5	0	5	7
12	12	n	7	0	3	4	7
12	13	n	7	0	7	3	7
12	14	n	8	0	0	4	8
12	15	n	7	0	7	3	8
12	16	n	7	0	5	4	8
12	17	n	8	8	7	2	8

Table B-15: Topic Feedback User 13

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
13	7	y	7	0	0	6	7
13	8						
13	9	n	7	9	0	9	8
13	10	n	9	10	0	2	8
13	11	y	5	7	0	6	6
13	12	n	3	0	5	9	3
13	13	y	5	0	4	9	3
13	14	y	9	0	0	1	8
13	15	n	9	0	0	5	7
13	16	n	0	0	9	2	8
13	17	n	8	10	9	2	6

Table B-16: Topic Feedback User 14

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
14	7	n	2	0	0	8	7
14	8	y	7	8	0	4	8
14	9	y	5	8	0	5	7
14	10	y	8	8	0	2	8
14	11	y	8	5	0	3	8
14	12	n	8	0	0	0	0
14	13	y	8	0	0	2	8
14	14	n	9	0	0	2	8
14	15	y	7	0	0	6	7
14	16	n	2	5	0	8	7
14	17	n	8	8	0	2	7

Table B-17: Topic Feedback User 15

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
15	7	n	2	4	1	10	8
15	8	n	0	10	0	2	10
15	9	n	0	10	0	0	10
15	10	n	7	7	0	3	10
15	11	n	7	7	0	4	10
15	12	n	5	0	2	3	10
15	13	n	8	0	0	1	10
15	14	y	7	0	0	1	10
15	15	y	10	0	0	2	10
15	16	y	7	0	9	7	9
15	17	n	0	0	0	0	0

Table B-18: Topic Feedback User 16

User Num	Topic Num	Did you know answer	Text	Object	Drawing	difficult	comfort
16	7	n	1	0	0	9	8
16	8	n	6	9	0	6	8
16	9	n	5	8	0	8	9
16	10	n	6	8	0	7	8
16	11	n	2	9	0	8	8
16	12	n	7	0	8	8	8
16	13	n	9	0	0	6	8
16	14	n	8	0	0	9	7
16	15	y	10	0	0	2	8
16	16	n	3	0	0	8	8
16	17	y	5	10	0	1	8