UNIVERSITY
of
GLASGOW

University of Koblenz-Landau
Institute for Computer Science
56016 Koblenz, Germany

Department of Computing Science
University of Glasgow
Glasgow G12 8RZ, United Kingdom

Interactive Video Retrieval

Diplomarbeit

by

Frank Hopfgartner

Supervisors: Prof. Dr. Steffen Staab, Institute for Computer Science, Computer Science Faculty
Dr. Joemon Jose, University of Glasgow

Glasgow, September 15, 2006

This thesis was submitted in partial fulfilment of the requirements for a Diplom-Informatiker degree at the University of Koblenz-Landau.

The author hereby declares that no third party was involved in writing this thesis. All sources and texts needed to do this work are listed in the bibliography.

. . . . . .
Frank Hopfgartner

An der Universität Koblenz-Landau eingereichte Diplomarbeit zur Erlangung des Grades eines Diplom-Informatikers im Studiengang Informatik.

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.
Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

. . . . . .
Frank Hopfgartner

Abstract

The goal of this thesis is to develop a video retrieval system that supports relevance feedback. One research approach of the thesis is to find out if a combination of implicit and explicit relevance feedback returns better retrieval results than a system using explicit feedback only. Another approach is to identify a model to weight existing feature categories. For this purpose, a state-of-the-art analysis is presented and two systems implemented, which run under the conditions of the international TRECVID workshop. It will be a basis system for further research approaches in the field of interactive video retrieval. Amongst others, it shall participate in the 2006 search task of the mentioned workshop.

Keywords:
Interactive Video Retrieval, Relevance Feedback, Query Expansion, TRECVID

Acknowledgements

> "No duty is more urgent than that of returning thanks [Schaff and Wace, 2002]."
>
> ST. AMBROSE
> *Bishop of Milan (340 – 397)*

This thesis was written during my stay as visiting research student at the University of Glasgow, United Kingdom. It was made possible within the framework of the European project K-Space.

Following Ambrose's advice, I want to take the opportunity to thank those persons who helped me, in different manners, with this thesis.

First of all, I am most grateful to my supervisor Prof. Steffen Staab, Head of the Research Group Information Systems and Semantic Web at the University of Koblenz-Landau. He enabled me to go abroad and supported me during the whole time. Thank you for giving me the opportunity to gain this fantastic experience.

I want to give thanks to Dr. Joemon Jose from the University of Glasgow for hosting and supervising me here. Thank you for your generosity, help and guidance in completing this thesis.

Moreover, I would like to thank the Glasgow Information Retrieval Group under Prof. Keith van Rijsbergen for the stimulating discussions we had and for making the trip a nice one. Especially, I want to express my gratitude to Jana Urban for her large contribution in this work.

My thank goes to all my fellow students and friends accompanying my path of life during the past years of study in several lectures, labs, term papers and, of course, everyday life.

My apologies to the others who I have not mentioned by name. I am indebted by them for the many ways they helped me.

Finally, I pay tribute to the constant support of my family, but especially of my parents. Without them, my whole studies would have been impossible and whose sacrifice, I can never repay. This one is for you!

Contents

1 Introduction

> "If I have seen further it is because I have stood on the shoulders of giants [Merton, 1993]."
>
> SIR ISAAC NEWTON
> *English mathematician (1754 – 1727)*

1.1 The history of videos

The invention of technologies to record and to show films and videos began centuries ago. In the 17th century, Christian Huygens, a Dutch physicist, realised an archaic movie projector called *Laterna Magica* (magic lantern). It was a very simple projector: with an oil lamp and lenses, images – painted on glass – could be projected onto a screen. But using this technique, it was only possible to display painted pictures. In 1839, Joseph Nicéphore Nièpce and Louis Jacques Mandé Daguerre invented the *daguerreotype*. It was a type of photograph, the image was exposed directly onto a polished surface of silver and iodine vapour. The pictures could be copied onto photo paper. In 1886, the Frenchman Louis Aimé Augustin Le Prince created a one lens camera which was capable of capturing movies. In November 1895, the brothers Max and Emil Skladanowsky started screening short films at the *Varieté Wintergarten* in Berlin. This was the start of the cinema which became a success especially in America. The cinema can be seen as the first great mass medium of the modern era [Faulstich, 2005].

An important step in the history of videos was the discovery of the analogue *television* as a system for broadcasting and receiving moving pictures and sounds which made it possible to reach even more people. Analogue television encodes television information by varying the voltages and/or frequencies of the signal.
The technical development went forward: In 1927, the film "The Jazz Singer" was screened which was the first feature-length motion picture with talking sequence [Abramson and Walitsch, 2003].

Colour TV began in the US on January 1, 1954. In the 1980's, the *Video Cassette Recorder* became popular. It uses magnetic tapes to record television broadcasts.

At the beginning of this millennium in the course of rapid societal transformation processes another new development in technology enters and consolidates an important position in the video business: The computers as multimedia equipment and other devices are going to change the handling of videos completely. Films are consistently broadcast, recorded and stored in *digital* form.
In 2003, Germany deactivated the entire old analogue broadcast signal in and around Berlin and now broadcasts only a digital signal. Digital television uses digital modulation and compression to broadcast video, audio and data signals. Other regions and countries will follow soon [Redaktion, 2003].
Grundig Intermedia informs, that the number of DVD Recorders sold in Western Europe increased by 400% from 2002 to 2004 while the distribution of Video Cassette Recorders decreased by 75% [Grundig Intermedia GmbH, 2005].

The more possibilities exist to store videos in a digital form, the more video files are archived. People are going to build their own digital libraries. Retrieval Systems have to be invented to assist the user in searching and finding video scenes he would like to see from many different video files.

1.2 Information Retrieval

Information Retrieval – or better Information Storage and Retrieval – is a summarising name for all methods to prepare, store and to find knowledge from data such as text documents. These three concepts are coherent as the preparation of data is done with regard to store them and to enable an easy retrieval [Luckhardt, 2006].
The importance of Information Retrieval has grown in the last few years. Web search engines such as `www.google.com` or `www.yahoo.com` are the most visible information retrieval applications. This year, Google Inc. even got ennobled as the Merriam-Webster Collegiate Dictionary now contains the verb *"to google"* in the meaning of using *"the Google search engine to obtain information ... on the World Wide Web." As in, "Let me google that."* as official English thesaurus [Chmielewski and Gaither, 2006].
According to Luckhardt [2006], in the stage of data preparation, it is common to detect the most important words or elements ("descriptors") from a document and to store them separately. This

proceeding is called *indexing*. The more expressive these descriptors are about the content of its documents the better. Ideally, the descriptors are catchwords. They can be selected manually or, more useful having a bigger amount of data, automatically. Therewith the list of automatic extracted descriptors does not contain needless terms, frequent words such as *and*, *not* or *with* can be filtered out. The list of these "non-descriptors" can be extended. For retrieving documents by their descriptors, both descriptors and document have to be concatenated. Such an index is called "inverted index". It is a matrix where each detected descriptor term has one row. Each column conforms one document. There, where column and row meet, is either a *1* if the documents contains the descriptor term or a *0* if not.
The inverted is normally stored in a database or in a file.
A retrieval engine takes search terms ("queries"), scans the inverted index for them and returns the columns matching the query. For focusing on specific topics, retrieval engines take advantage of the boolean algebra. This enables the option to specify search queries using more than one term, e.g.
Glasgow AND Koblenz will return columns matching both terms,
Glasgow OR Koblenz will return columns matching one (or both) terms and
Glasgow AND NOT Koblenz will return columns matching only the first term.

The first retrieval cycle in an information retrieval process does not always provide satisfying results [Campbell, 2000b]. There are various reasons for this: the terms might just not appear in the document, the user tried unfavourable terms or he does not know meaningful terms and was not specific enough.
Hence it is necessary to refine a query – to *expand* it. This can happen manually in adding a new term or automatically. For improving an automatic query expansion, user give *feedback* on the relevance of items [Rocchio, 1971]. This means that he judges the relevance of already retrieved documents and hence signifies the direction he wants to specify. Based on user feedback, system can support new terms for query expansion.

1.3 Scope of the Thesis

The aim of this thesis is to develop an interactive video retrieval system. The need for such a system has been introduced in chapter 1.1. Besides, the Information Retrieval Group at the University of Glasgow this year participates for the first time in TRECVID, a workshop with focus on video retrieval (read more about it in chapter 6). The research group has a main focus on information retrieval. However, a functioning video retrieval system did not exist yet, as video retrieval is a quite new field of study in Glasgow. The now in the scope of this thesis developed software can

be used as a basic system for participating in the workshop. Its development was oriented on the guidelines of TRECVID and tested using the provided 2005 data set.
Part of the development was the examination of information retrieval techniques. A short introduction has been given in chapter 1.2.

Following this short historical introduction and set of technical overview to the subject, this thesis' tasks will be presented in the following way:

Chapter 2 – Video Data Processing
In this chapter the scientific approaches are presented that are relevant in the scope of this thesis. It starts with a description of the MPEG-1 video format in chapter 2.1, the coding format for the files of the relevant test set for this thesis. In chapter 2.2, the need and the technique for shot boundary detection is explained. After separating a video into different shots, every shot can be treated like an independent part of a video. This is adaptable in particular concerning news in a television broadcast where – in an ideal scenario– every shot discusses a new topic. In chapter 2.3, the Automatic Keyframe Extraction is explained which is useful for a later – content-based – presentation of the detected shots.
Chapter 3 – Video Search
In this chapter, the main challenges in video search are introduced: Chapter 3.1 explains the difficulties in dealing with the gap between low-level content that can be computed automatically and the subjectivity of semantics in high-level human interpretations. In chapter 3.2, the semantic visual feature ontology is presented. Chapter 3.3 introduces relevance feedback techniques which are useful to bridge this gap. Chapter 3.4 gives a survey on how automatic query expansion can help users in finding the right results.
Chapter 4 – Discussion
This chapter bears a critical discussion on the different technologies that have been presented in the previous chapters. It starts with an argumentation about the best shot boundary detection method in chapter 4.1. Chapter 4.2 deals with the most optimised extraction of the keyframes while the chapters 4.3 and 4.4 contain a discussion on the feature extraction and the relevance feedback respectively. Chapter 4.5 discusses the benefits of interactive versus automatic query expansion.
Chapter 5 – Interactive Video Retrieval
After introducing the main features in video retrieval it is mandatory to have a closer look on the idea of retrieval engines. This chapter gives a survey about interactive video retrieval. Chapter 5.1 explains the concept of video surrogates. In chapter 5.2, the problem of representing videos for browsing using *good* keyframes or fast forward techniques is introduced. In chapter 5.3, a survey is given on video indexing methods. Chapter 5.4 provides an overview on users' video relevance criteria. In chapter 5.5, the most important approaches are presented.

Chapter 6 – TRECVID
To reach comparable research in any scientific sector, it is necessary to provide scientists a broader platform for timed presentation and alteration of their work, especially in an advanced and continuously changing sector. One of these platforms is the TRECVID workshop which is presented in chapter 6.1. The organisers offer a data set to all participants which is described in chapter 6.2. In chapter 6.3, some systems developed by participants of past TRECVID workshops are presented to give an insight and overview of recent developments. To provide comparison between the efficiency of these systems, TRECVID creates search tasks. These search tasks are described in chapter 6.4.
Chapter 7 – Software Design
The Software Crisis in the late 1960's [Dijkstra, 1972] leaded to a reflecting how to develop and implement software tools. Computer programs which have been programmed without any documentation became a bigger problem as it was not easy or even impossible to continue or correct them. This was the hour of birth for Software Engineering which utilises the design, use and further development of software systems. Software systems consist of source code and its accompanying documents which are useful and helpful for the usage of the program. Different approaches how to proceed in developing a software system have been introduced. The design of this system is oriented on the Object-Oriented Analysis and Design (OOAD) by Booch [1995]. The process covered with this chapter is divided into a requirements analysis in chapter 7.1 followed by a presentation of use cases and its scenarios in chapter 7.2.
Chapter 8 – Implementation and Documentation
A good documentation of a developed software system is mandatory, as it helps others in understanding the structure and the source code of the system. This chapter offers a closer look at the structure of the software. After a short overview in chapter 8.1, chapter 8.2 explains the requirements and infrastructure for the system here at Glasgow University. Chapter 8.3 presents more technical details about the developed parser while chapter 8.4 presents details about the multimedia retrieval tool.
Chapter 9 – Evaluation
The developed system can be the base for various research in the field of video retrieval. One research question is presented in chapter 9.1. Chapter 9.2 presents the result of a simulated user study. Chapter 9.3 explains the setting for a TRECVID user study. Questionnaires for evaluation are introduced in chapter 9.4. Chapter 9.5 explains the common experimental procedure. **Chapter 10 – Conclusion and Future Work**
Giving a final reflection on the finished work, this chapter draws a conclusion in summarising its cognitions and illustrates the course of the work in chapter 10.1. Chapter 10.2 summarises the findings of this thesis. In chapter 10.3, final remarks point to ideas and approaches that have not

been considered in the developed software system but that are worth being focused on in a future work.

2 Video Data Processing

"Divide and conquer"

Important algorithm design paradigm in Computing Science

In this chapter the scientific approaches are presented that are relevant in the scope of this thesis. It starts with a description of the MPEG-1 video format in chapter 2.1, the coding format for the files of the relevant test set for this thesis. In chapter 2.2, the need and the technique for shot boundary detection is explained. After separating a video into different shots, every shot can be treated like an independent part of a video. This is adaptable in particular concerning news in a television broadcast where – in an ideal scenario– every shot discusses a new topic. In chapter 2.3, the Automatic Keyframe Extraction is explained which is useful for a later – content-based – presentation of the detected shots.

2.1 MPEG-1

MPEG-1 (also called ISO/IEC 11172) is a standard released by the Moving Picture Experts Group (MPEG). It was their first standard, others followed later. The standard supports the coding of audio and video in a container format at a bit rate of up to 1.5Mbps. The quality of MPEG-1 encoded videos is not acceptable for consumer viewing but for processing, previewing and analysing videos it is adequate. Newer standards like MPEG-2 are more useful for consuming. Videos are a series of individual frames or frames displayed at a constant rate. The MPEG-1 standard encodes its videos with a frame rate of 25 frames per second. It achieves a high compression rate by the use of motion estimation and its compensation between frames. It uses the fact that there are little changes in the picture from frame to frame. There are usually only little movements of single objects apart from changes in a scene or shot. So MPEG divides a frame into different macroblocks which can be compared across frames. If a macroblock

appears on another frame (either because it has not moved in time or has moved in some direction by only a small amount), it is not encoded entirely in the following frame. Instead, the difference between the two macroblocks and their motion vector is encoded [Watkinson, 2001].
Many different algorithms can be used to detect the best macroblock. A search in the course of encoding brings the best results but it is also computationally expensive. Alternatively, one can use a logarithmic search, one-at-a-time search, three-step search and the hierarchical search. These techniques are described in [Gong et al., 1996]. The appropriate search algorithm used is subject to the encoder.

2.2 Shot Boundary Detection

Coupled with the increased power of computing, content-based manipulation of digital videos is now increasing. To afford content-based navigation in a video, it is necessary to break up the data into structured elements. In the case of video, these elements are *shots* and *scenes*.
A shot is defined as a part of the video that results from one continuous recording by a single camera. A scene is composed of a number of shots, while a television broadcast consists of a collection of scenes. The gap between two shots is called a shot boundary. According to Zhang et al. [1993], there are mainly four different types of common shot boundaries within shots:

- *A cut*: It is a hard boundary or clear cut which appears by a complete shot over a span of two serial frames. It is mainly used in live transmissions.
- *A fade*: Two different kinds of fades are used: The fade-in and the fade-out. The fade-out emerges when the image fades to a black screen or a dot. The fade-in appears when the image is displayed from a black image. Both effects last a few frames.
- *A dissolve*: It is a synchronous occurrence of a fade-in and a fade-out. The two effects are layered for a fixed period of time e.g. 0.5 seconds (12 frames). It is mainly used in live in-studio transmissions.
- *A wipe*: This is a virtual line going across the screen clearing the old scene and displaying a new scene. It also occurs over more frames. It is commonly used in films such as *Star Wars* and TV shows.

As these effects exist, shot boundary detection is a non-trivial task. It is not known before, when these effects will appear.

There have been a number of various approaches to handle different shot boundaries, including

calculating pixel differences between neighbouring frames, macroblock comparison from MPEG-encoding, comparison of neighbouring frames using colour-histograms and the comparison of edges in frames. All approaches work well for different transition types but cannot be used for every shot boundary. Frame comparison based on colours for instance works fine on cuts but does not detect dissolves or fades. Edge detection works effectively in wipe and dissolve detection.

Main research on this topic is promoted at Dublin City University, Ireland. Their Centre for Digital Video Processing is developing a digital video library called Físchlár which uses shot boundary detection as the very core of video indexing. According to Smeaton et al. [1999], Browne et al. [2000], the following sections cover different approaches that have been implemented and evaluated by this university while designing their system.

2.2.1 Shot Boundary Detection Based on Colour Diagrams

The first approach tested at Dublin was a *shot detection based on colour histograms*. They computed frame-to-frame similarities based on colours which appeared within them, albeit of the relative positions of those colours in the frame. After computing the inter-frame similarities, a threshold can be used to indicate shot boundaries. A detailed description on that attempt can be found in [O'Toole, 1999]. More research in this approach has shown that a colour-based detection has no good threshold [Smeaton et al., 1999]. It needs dynamic thresholding to work on other effects than simple shot boundaries.

2.2.2 Edge Detection

The next approach is *Edge Detection* which is based on detecting edges in two neighbouring images and comparing these images. It should be possible to detect all kinds of shot boundaries by detecting the appearance of edges in a frame which are far away from the ones in the previous frame. The tested approach in Dublin used over 2 hours and 40 minutes of video files of different TV broadcasts [Smeaton et al., 1999]. They spotted various reasons why their programme missed a real cut between scenes:

- blurred images where the edges could not be defined clearly
- images with similar backgrounds or intensity edges to the next-following image
- dark or bright images where the edges are not defined in an accurate manner
- straight cuts from a blank screen to a dark screen

- a cut between different camera perspectives showing the same scene

They also detected reasons for wrong identification of cuts:

- fast action scenes with fast moving and changing edges
- camera flashes
- close-up moving scenes
- objects moving in front of the camera lens without being present on the image before
- a zoom out or in, camera pan or any camera motion
- computer generated scenes
- interferences in the video from broadcasting or recording
- an object cut from an image

Main problems for missing cuts in all kinds of videos are cuts between dark scenes and the detection of so-called pseudo-cuts during the credits at the end of a film or programme.
They also found out that the detection of false shots increases with the quality and size of the example videos. Since many false detection had occurred because of camera panning and/or zooming [Smeaton et al., 1999] they implemented a technique to compensate these movements. This solution can counter problems caused by dissolves and fades and other changes using soft colour changes. The advantage – compared to colour based shot detection – is that this technique will not be fooled by colour changing effects like a flash. But on the other side, each frame has to be decoded, so it runs very slowly.

2.2.3 Shot Boundary Detection Using Macroblocks

Besides, they investigated the *shot boundary detection using macroblocks*. Depending on the types of the macroblock the MPEG pictures have different attributes corresponding to the macroblock. Macroblock types can be divided into forward prediction, backward prediction or no prediction at all. The classification of different blocks happens while encoding the video file based on the motion estimation and efficiency of the encoding. If a frame contains backward predicted blocks and suddenly does not have any, it could mean that the following frame has changed drastically which would point to a cut. This approach, however, becomes difficult to implement when there is a shot change, and the frame in the next shot contains similar blocks as the frame before.

2.2.4 Evaluation

To evaluate the different attempts, the researchers tested all of them on the same data set. For a useful evaluation, they had to consider the number of false shot boundaries detected by the method, the number of shot boundaries not detected by the method and the number of actual shot boundaries in the baseline. As estimated before, a good number of shots where detected by the different techniques. However, many shots were only detected by one of the three methods or by none of the techniques.

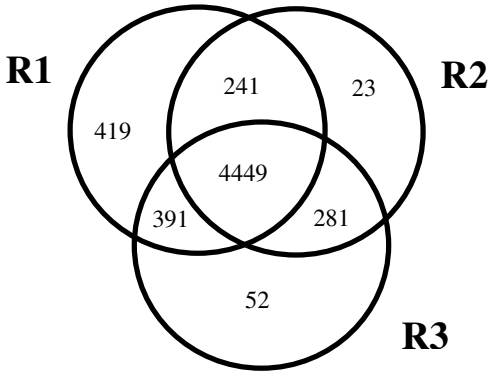

Figure 2.1: Overlap in correct shot boundaries detected by the methods over the complete corpus [Browne et al., 2000]

Figure 2.1 illustrates their outcome. R1 is the colour histogram approach, R2 is the Edge Detection while R3 is the Macroblock method. R2 and R3 are not as successful as R1 but their cumulated result adds another 356 correct shots to the result of R1. Therefore, it makes sense to concentrate not only on one technique but to use all techniques [Browne et al., 2000].

2.2.5 Combining Shot Boundary Detection Algorithms

In his satiric novel *Candide*, French philosopher Voltaire formed the saying of using the *best of both worlds* [Voltaire, 1984]. It is generally applied when it is better to use two alternatives in parallel instead of selecting one alternative to benefit from both advantages. As shown before, the different attempts for shot boundary detection are worse or better for the different kinds of shot boundaries. Following Voltaire, only a combination of all approaches could bring the best results. In [Browne et al., 2000], the researchers compared these methods and decided to use a weighted boolean logic to combine the different approaches. Their attempt favours the results of the colour histogram method which gives best results in terms of performance. If the difference value is,

however, above a specific low value, it will select one of the other methods. On their test data, the combined approach had a negligible effect on precision for all programs but the news reports.

2.3 Automatic Keyframe Extraction

As the final goal is to assist the user in finding specific shots about topics he is interested in, visual tools are needed that help users find the information they are looking for. To be more specific, these visual tools should provide multi-point access to the linear, time-based medium of the video. For practical purposes, information that describes the content of the recorded video best should be extracted. This representing information is called keyframe. Keyframes can be used as a kind of visual index or thumbnail image while using a Graphical User Interface. Figure 2.2 illustrates this procedure of shot boundary detection and keyframe identification.

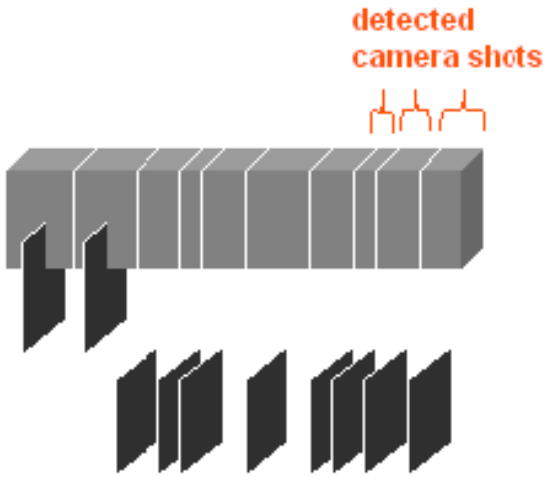

Figure 2.2: Shot boundary detection and keyframe identification [Smeaton, 2002]

One possible and simple solution to detect keyframes is to take any frame e.g. the first or the middle one as a keyframe. Although, this is a kind of random treatment in detecting keyframes, it is used by most shot boundary detection systems in literature [Browne et al., 2000]. However, the frames should previously be analysed to extract keyframes that really represent the content of the shot. Ideally, the best keyframes would be those which are aligned to the users' wishes. As an example, a user might ask for a keyframe of a news report about elections for the Scottish Parliament which contains the political candidates. Detecting these specific keyframes needs long

term content analysis in shots.

Philips Research investigated in this approach and developed a system called Vitamin [Dimitrova et al., 1997]. While detecting different cuts in a video, a huge amount of frames is extracted. This can be thousands of frames for one hour of video. Though, not all frames are important or convey finding the content of the shot. That means that they use filters to minimise useless frames: They filter out *blank* (unicolour) frames or recurrent frames. For instance in a dialogue, both speakers will be shown very often. Only two keyframes, one for every speaker, will be considered for this scene. They take all frames of a shot and divide them into different blocks that consist of various regions. As a frame consists of different regions, similar frames can therefore be chosen by comparing their blocks and regions. Summarising all these aspects, they state that all frames are also accepted as keyframes if they are not too similar to a previous frame that was selected as keyframe.

3 Video Search

> "We are drowning in information, but starving for knowledge [Naisbitt, 1982]."
>
> JOHN NAISBITT
> *American writer (born 1929)*

In this chapter, the main challenges in video search are introduced: Chapter 3.1 explains the difficulties in dealing with the gap between low-level content that can be computed automatically and the subjectivity of semantics in high-level human interpretations. In chapter 3.2, the semantic visual feature ontology is presented. Chapter 3.3 introduces relevance feedback techniques which are useful to bridge this gap. Chapter 3.4 gives a survey on how automatic query expansion can help users in finding the right results.

3.1 Features to bridge the Semantic Gap

The universe of the American TV show Star Trek is a paradise of information retrieval backed by computers. Computers are everywhere and can do nearly everything to provide viewers an increasingly realistic image of a world captured by modern media. The survival of Captain Picards starship Enterprise is due to its perfect, heroic crew and, above all, to the incredible computer system. It is the computer and its abilities, that make Star Trek seem like a show about the future of mankind backing our fantasy with modern technology. But are these systems achievable, are the tasks they perform possible? The screenplay writers' vision of the information technology's future appears quite realistic: Besides technological improvements like the voice interface, they introduced a system that has further knowledge about the content of data. It brings on so called content-based retrieval.
Now in the 21st century, today's computer systems are not that powerful. One of the main issues that have to be solved in content-based retrieval is called the Semantic Gap. This is the difference

of information between low-level data representation and high-level concepts which the user associates with retrieved data [Urban and Jose, 2004].
To bridge this gap in multimedia retrieval, different techniques have been developed for visual feature extraction. Following Zhao and Grosky [2002] and Souvannavong et al. [2004], integrated features are *colour, texture, shape, motion* and *text*.

- *Colour* is one of the most commonly used visual feature in image retrieval as it is relatively simple to extract. More precisely, the colour histogram method is the mainly used representation method. It statistically describes the combined probabilistic property of the colour channels e.g. red, green and blue.
- The *textures* deal with the patterns in an image presenting the properties of similarity that do not result from the dominance of a single colour or intensity value. Three different categories of texture-based techniques exist: the *statistical, spectral,* and the *structural approach*.
- The *shape* representation can be divided into *boundary-based* which uses only the outer boundary characteristics of the entities and *region-based* which uses the entire region. The feature is useful as it is invariant to translation, rotation and scaling.
- The *motion* feature is one of the most effective approaches. It is useful to extract activities in a video shot. Two different motion features are selected: the *motion histogram* of a shot or the *camera motion*.
- Declared as to be very important is also the *text feature*. It can help finding the semantic content by providing information from automatic speech recognition. Text is not considered as low-level feature though.

These extracted so called low-level descriptors can be used as an input for the extraction of higher level information. Besides, they can be used for similarity matching based on the descriptors. A segmentation usually starts with shot boundary detection (see section 2.2) over segmentation on a semantic level like scene segmentation. In accordance to Bailer et al. [2005], some feature extraction approaches have been implemented and evaluated for extracting higher level information. The paper gives an overview of the state-of-the-art technologies:

- *Motion Segmentation:* For the object recognition, a segmentation of regions is a key step. An object can consist of different colours or shapes but its motion is the subject. So it is capable to segment semantically meaningful regions.
- *Video OCR:* Video OCR is a special challenge as it is more difficult than pure text OCR. There usually is a lower resolution, additionally complex backgrounds and the text some-

times appears in a different slant. A video OCR tool consists of three steps: Text Detection, Text Segmentation and Enhancement and Classical OCR.

- *Automatic Speech Recognition:* A fine feature extraction method is the automatic speech recognition. It refers to multiple cross-knowledge and application domains like acoustic, phonetics, linguistic and lexical domains. An Automatic Speech Recognition System contains different approaches. The main approaches are: Audio Segmentation, Speaker Segmentation, Speaker Identification and Speech Transcription.
- *Face Detection and Recognition:* The challenge in Face Detection is to find regions in arbitrary sized images that contain a human face. The problem is that faces may appear in different scales, rotations and head poses. An aggravating factor is the background which might be complex or the different illumination. As faces are non-rigid objects, a lot of variations are possible. An interesting approach is the Físchlár-Simpsons Video Retrieval System developed at DCU. They implemented a system for Face Detection in the famous American TV show "The Simpons" [Browne and Smeaton, 2005].
- *Event Detection:* The major coverage in Event Detection is in the sports area. A prior knowledge about audiovisual features that appear in a sports game is necessary (e.g. the fans will cheer after a foul in football). In Event Detection, it is necessary to concentrate both on the video and the audio material as they are associated. A good overview of the different approaches in the sports field can be found in [Adami et al., 2003].
  Another event is a dialogue scene. There are different approaches dealing with the detection of dialogue scenes between two or more speakers. An overview can be found in [Haberfehlner, 2004].

3.2 Semantic visual feature ontology

To facilitate video retrieval, it is a good approach to define a high level semantic description of video content, a so-called semantic visual feature. In fact, a lot of research has been done on incorporating semantic concepts with visual data [Koskela et al., 2006]. The aim is to enrich traditional example-based retrieval via relevance feedback with semantic concept models. These models have to be trained off-line with training data.
In [Naphade et al., 2005], the authors proposed a 39-feature lightweight ontology to break down the semantic space. Their ontology – called LSCOM-Lite – has two layers. The upper layer consists of seven categories: Program Category, Setting/Scene/Site, People, Objects, Activities, Events and

Graphics. The secondary layer consists of sub-categories to offer further classification. Table 3.1 shows one example category including its sub-categories.

This categorisation enables the user to find similar shots easily by browsing to shots that have similar visual features like a selected shot. An example: The user wants to find shots that show the face of Tony Blair. This shot is classified to *politics, face, person, government, leader* and *police/private security personnel*. It would be useful to list more results matching these features.
At the University of Wisconsin-Milwaukee, researchers tested this approach. They implemented a system and ran a user study. The test included two different types of video searching tasks: *Visual centric tasks* (VCT) with particular focus on visual features of the keyframes and *non-visual centric tasks* (NCT) with focus on non-visual features of the keyframes. The test obtained that the Semantic Visual Feature was very effective for the VCT tasks, but not for the NCT tasks [Mu, 2006].

3.3 Relevance Feedback

For state-of-the-art retrieval systems, it is rarely possible to retrieve relevant complex results in the first iteration [Campbell, 2000b]. Very often, the original search query has to be modified, completed or changed entirely. Thereby, retrieved results can serve as a new source to adjust the query. In Multimedia Retrieval, this adjustment can both be based on the low- and high-level content presented in chapter 3.1 and the categorical semantic ontologies from chapter 3.2.
However, to decide, which feature or which ontology is relevant for the current search, the system needs a feedback from the user, so-called Relevance Feedback. The idea of including relevance feedback to a retrieval system was first researched for text retrieval systems [Rocchio, 1971].
The iterative process of the query-based systems usually consists of the following states:

1. The system lists retrieved results after processing a search query.

- Program Category
    1. Politics: Shots about domestic or international politics
    2. Finance/Business: Shots about finance/business/commerce
    3. Science/Technology: Shots about science and technology
    4. Sports: Shots depicting any sport in action
    5. Entertainment: Shots depicting any entertainment segment in action
    6. Weather: Shots depicting any weather related news or bulletin
    7. Commercial/Advertisement: Shots of Advertisement, commercials

Table 3.1: Example of Semantic Visual Feature Ontology

2. The user provides a feedback to the system e.g. in rating the relevance of a result.
3. The system updates the retrieved result list.

Thereby, it depends on the user how many iterations are necessary until a satisfying result is retrieved. An example is given in figure 3.1.

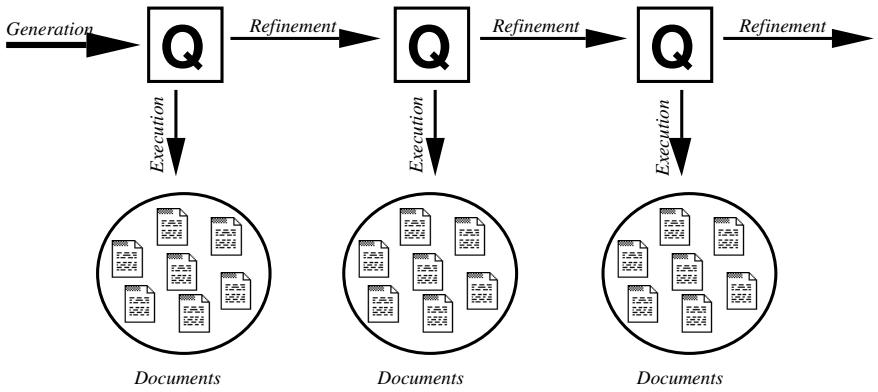

Figure 3.1: The iterative process of query-based systems. [Campbell, 2000b]

The quality of the results also depends on the ability of the system to improve its retrieval. A characteristic of existing systems is *how* they react on the user's behaviour. In this case, one can distinguish between two approaches: explicit and implicit relevance feedback.

The *explicit relevance feedback* assumed in most current systems like [Cox et al., 2000, Porkaew et al., 1999, Tong and Chang, 2001] asks the user to rate the relevance of a retrieved document. The user interface of course has to provide a possibility to input this judgement by the user. As it is not always easy or feasible for the user to judge the relevance of a document, this task is often seen as a burden. In addition, one problem is, that the user does not always want to mark the relevance of the documents, as it means extra work for him [Xu, 1997].

A less-distracting way to gain feedback is the *implicit relevance feedback*. In this case, the system observes the user's interaction and automatically rates the documents he accessed before. The advantage of this is that the user does not have to mark the relevance of retrieved documents. A disadvantage is the quality of the results which are implicitly marked as relevant: The information is not as adequate and clear as from *explicit* feedback. However, implicit feedback can be seen as an effective substitute for its explicit counterpart [White et al., 2002].

The relevant feedback approach relies on a quite simplified view of the real world. It assumes that the user's information need is static. There is no need to update the user's judgements. However, the user's behaviour is more chaotic. His *actions* are time dependent which is the result of giving

inconsistent feedback and, even more importantly, the *goals* the user wants to reach are also time dependent. His search goal might change either gradually or even abruptly as he gets new ideas and influences from retrieved results heading into another direction as originally. To handle this non-static behaviour, Campbell and van Rijsbergen [1996] proposed the *Ostensive Model*, which covers *"the intentionality of an information need that is assumed to be developing during the searching session"* [Campbell, 2000a].

In this model, the search query is not only evolved on one chosen document which is rated as relevant, but is dependent on the path of documents the user passed in his search session. As the user's goal of the search might change while the search itself, the model adds a temporal dimension to the notion of relevance. Recently added documents are declared more important than older results, as they should be closer to the result the user wants to achieve [Campbell, 2000b].

Gurrin et al. [2006] evaluated methods of relevance feedback for video retrieval engines working with TV news data. They identified an optimal number of terms to compose a new query for feedback. They also analysed that the number of documents do not have a great effect on the optimal number of terms. They concluded "*that for shots a system will perform at or near its peak when 7-8 terms are used to generate a new feedback query and for TV news stories that the peak can be found in most cases when 10-13 terms comprise the query"*.

3.4 Query Expansion

The original, manually entered query is most important as there are many different ways to describe the same object or event. However, it is nearly impossible to formulate a perfect query at first attempt due to the uncertainty about the information need and lack of understanding on the retrieval system and collection. The original query indicated what the searcher really wants, but a problem is, that a query might not be precise enough or that a retrieval misses videos that have semantic similarities but no speech similarities. For instance, if the user enters "George W. Bush", the results will miss keywords like "President of the U.S." or "Governor of Texas" in the ASR transcript. However, some results might refer to the plant "bush" which is not relevant in this case. So, there is the need to find a way to expand a query such that the redefined query better fits the target topics and brings on more relevant results [Zhai et al., 2006]. A simple way to do so is to use relevance information from the user. The content of the relevant-rated documents can be used to form a new, expanded query expression which is ranked by some measures that describe how useful its terms might be [Robertson and Spärck Jones, 1990]. Dependent how much influence the user shall have, the expansion terms can either be added by the user – *interactive query expansion*

– or by the system – *automatic query expansion*.

Different query expansion techniques have been tested, e.g. [Beaulieu, 1997, Efthimiadis, 1996]. In [Zhai et al., 2006], the authors propose an *automatic* query expansion technique. It expands the original query to cover more potential relevant shots. The expansion is based on an automatic speech recognition text associated to the video shots. After triggering a first retrieval using the query Q_{i-1}, the user can rate a set of shots as relevant and another set which is rated irrelevant. They are denoted as positive D^+ and negative D^- sets. Based on the positive set, a keyword histogram $WH_{D^+} = \{({a_1}^+, {W_1}^+), ({a_2}^+, {W_2}^+), ..., ({a_m}^+, {W_m}^+)\}$ is computed, where ${W_1}^+$ is the extracted keyword accompanied by its normalised frequency ${a_1}^+$ in the positive set. Another histogram based on the negative set is developed similarly: $WH_{D^-} = \{({a_1}^-, {W_1}^-), ({a_2}^-, {W_2}^-), ..., ({a_m}^-, {W_m}^-)\}$. When starting a new search, the new query $Q_i = WH_{D^+}$ is submitted to the retrieval engine. In this step, the query is expanded (from Q_{i-1}) to a larger set Q_i. The relevance of a retrieved shot is calculated by computing the histogram correlations. Dependent on a given shot S, a normalised keyframe histogram WH_S is calculated as vector product, $R(S) = VP(WH_S, WH_{D^+}) - VP(WH_S, WH_{D^-})$. $VP(.,.)$ represents the inner product of the vectors. The vectors WH_S, WH_{D^+} and WH_{D^-} are restructured by filling the missing positions with zeros to have the same dimension.

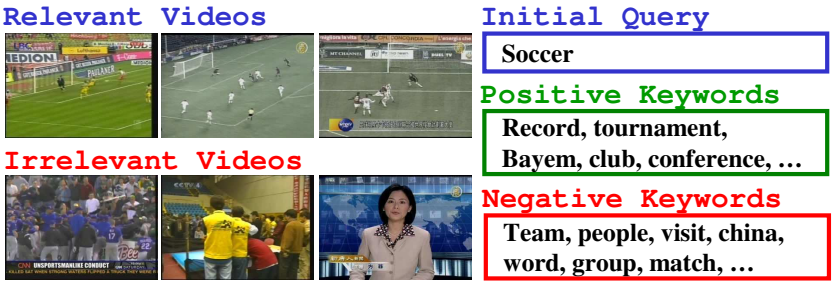

Figure 3.2: Example for the automatic query expansion [Zhai et al., 2006]

Figure 3.2 shows an example for the automatic query expansion. The original search query is *soccer*. The figure shows the examples of the videos rated *relevant* and rated *not relevant* and the positive and negative keyword sets.

Another approach – the *interactive* query expansion – is discussed e.g. in [Magennis and van Rijsbergen, 1997]. The idea is that the automatically-derived terms are offered as suggestions to the searcher, who decides which to add.

A variant is the so-called *Pseudo-relevance* or *local feedback* [Xu and Croft, 1996]. It is assumed that the top ranked documents retrieved after a first cycle are relevant. They are automatically

marked as relevant – others maybe as non-relevant – and the query automatically expanded. Using this expanded query, another retrieval can be done. The technique was first introduced by Attar and Fraenkel [1977]. In this paper, top-ranked results for a query were proposed source of information for detecting new query terms.

4 Discussion

> "It is better to debate a question without settling it than to settle a question without debating it [Joubert and Auster, 1983]."
>
> JOSEPH JOUBERT
> *French moralist and essayist*
> *(1754–1824)*

This chapter bears a critical discussion on the different technologies that have been presented in the previous chapters. It starts with an argumentation about the best shot boundary detection method in chapter 4.1. Chapter 4.2 deals with the most optimised extraction of the keyframes while the chapters 4.3 and 4.4 contain a discussion on the feature extraction and the relevance feedback respectively. Chapter 4.5 discusses the benefits of interactive versus automatic query expansion.

4.1 Shot boundary detection

In text retrieval, documents are treated as units for the purpose of retrieval. So, a search returns a number of retrieved results. It is easy to design a system that retrieves all documents containing a particular word. The user can browse through the results easily to find parts of interest. If documents are too long, techniques have been developed to concentrate on the relevant sections [Salton et al., 1993].

This practice cannot be used for videos. If videos are treated as units of retrieval, it will not lead to a satisfactory result. After relevant videos have been retrieved, it is still an issue to find the relevant clip in the video. Especially as most clips have a duration of only a few seconds. Even if these small clips are seen as associated stories of several minutes of length, it is not optimal. It is time consuming to browse through all video *sections* to find the relevant part [Girgensohn et al., 2005]. Visual structures such as colour, shape and texture can be used for detecting shot boundaries and

for selecting keyframes [Aigrain et al., 1996].

However, seperating videos into different shots is not the best solution as the context of a shot is not often clear. Very often, a shot is only understandable when it is played in its context. A shot e.g. showing a public square full of people waving flags shows nothing more than a crowded square. Seen in its context, this crowd might be celebrating a victory of their favourite football team, celebrating the national holiday or demonstrating against something. Keeping the context of a video part is important for understanding it.

4.2 Keyframe Extraction

According to Yang and Marchionini [2005], current automatic keyframe techniques as presented in chapter 2.3 are good in selecting unalike keyframes for representing shots in a video. However, all methods focus on physical attributes of the video frames and not on the users' understanding and intention. Ideal keyframes which represent a video shot should afford the users several cues to build visual gist. Their user study demonstrated that users can be highly effective in identifying visual features to make sense of a video.

4.3 Feature Extraction

To this day, there has been no serious research which low-level detectors can be used to identify which kind of images. Dr. Xavier Hilaire from Glasgow University is working on that issue.
It is noticeable that the *colour* feature (dominant colour) could be useful to detect natural landscapes like green grassland or a beach with a blue sky. *Textures* could be useful to identify natural material like clothes. The *shape* feature might mainly be useful to identify single objects in a picture such as a helicopter in the sky. Searching for *motion* can help detecting moving objects such as aeroplanes taking off. However, searching for a static picture of a skyline might also be found when retrieving for motion, as the traffic on the street causes motion.
One approach has been worked on at the University of California, Berkeley. They built statistical models to *explain* the data in a collection. Once a model has been built, it can be queried. In their system called Blobworld [Carson et al., 1999], they built the models in grouping pixels into regions by modeling the distribution of colour, texture and position frames. After grouping, the regions are described using colour and texture properties. Finally, they store these models and use them to retrieve similar images. Figure 4.1 shows an example representation in Blobworld.

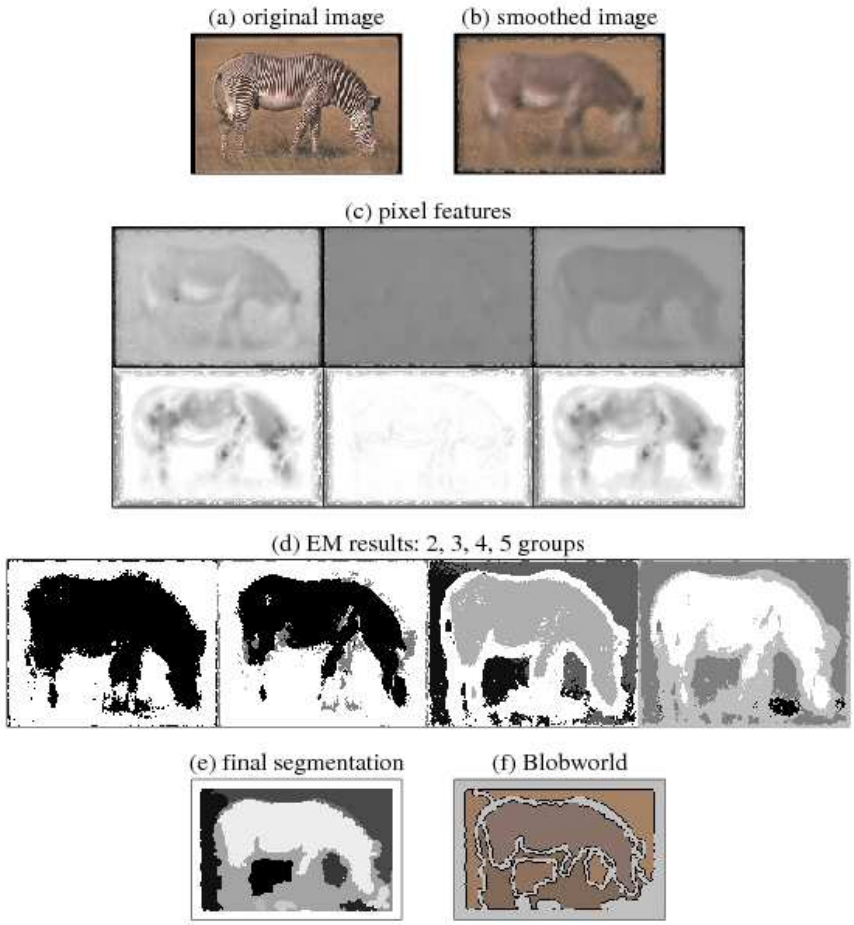

Figure 4.1: Creating the Blobworld representation. [Carson et al., 1999]

Even though this is an interesting approach, it is not an effective solution as it is necessary to create a model for every object, which is an enormous effort.

4.4 Relevance Feedback

According to Campbell [2000b], weighted-term systems can supplement search queries by incorportating relevance feedback. The content of relevant-marked documents might be a richer source of information for retrieval than the initial query. Nevertheless, it changes the role of operation and interaction of a query-based system to a user.
Campbell [2000b] compared the techniques of manual query modification versus using a relevance feedback technique:
Initial queries probably produce only one or two results of very weak relevance. Experiments

suggest to use the initial query, to identify the relevant results and to rate them. Manual query modification would need several retrievals until any *good* documents are retrieved. Using the relevance feedback technique, each iteration could output results that are increasingly relevant because of the more relevant-rated documents which contain richer information for the system.
A manual query modification has more the touch of a try-and-error-method while using relevance feedback, the user can have the feeling to get closer and closer with each new retrieval he triggers. If a user tries to modify his query, he must not only have the knowledge of the subject matter to find relevant words, but also should be able to find *effective* terms. Effective terms are words that are infrequent in the whole data collection but frequent in the targeted documents. On the other side, in using relevance feedback, the user identifies results which are, to some extent, relevant.
Many developed video retrieval systems such as the Informedia system from CMU (see chapter 6.3.1) or the Físchlár Digital Video system (see chapter 6.3.2) support single video document feedback, but also relevance feedback where more than one video can be marked as feedback. Conventional relevance feedback techniques can select terms and image features for query refinement [Gurrin et al., 2006].

4.5 Query Expansion

Considering query expansion it is important to find out which technique is more useful – *interactive query expansion (IQE)* or *automatic query expansion (AQE)*.
According to Ruthven [2003], the main argument for prefering AQE is that the system can take advantage of using more statistical information to acquire the relative utility of expansion terms. Hence, it can make a better selection which terms to take into account. The main argument for IQE is that it gives more control to the user. As the user decides which criterias to take for relevance, he should also be able to make a decision on which terms could be useful [Koenemann and Belkin, 1996].

Several user studies have been done to find out the relative merits of AQE versus IQE.
Koenemann and Belkin [1996] showed that IQE shows a better performance than AQE for specific tasks, while Beaulieu [1997] demonstrated that AQE gives higher retrieval effectiveness in an operational environment. The difference of their results can partly be explained by the different interfaces they used. Also search tasks and experimental methodology can effect the results.
Magennis and van Rijsbergen [1997] measured the effectiveness of IQE in live and simulated user experiments. There, they estimated the best performance for making IQE decisions. They concluded that users tend to make sub-optimal decisions on how to expand the query.
Ruthven [2003] investigated the potential effectiveness of IQE. He made a user experiment to

supplement experimental investigations of IQE decision-making. The results showed that IQE can be an effective technique compared to AQE. He claims that the idea that the user may be better in identifying good expansion terms than the system might be partly true for certain types of retrieval. Fowkes and Beaulieu [2002] analysed that users prefer IQE when dealing with complex query statements. They may also be more competent in targeting specific aspects of a retrieval like focusing on parts of the information. Ruthven's final conclusion is that it is not easy to achieve what are the potential benefits of IQE. His results show that users in particular have difficulties in identifying useful expansion terms. This implies that simple interfaces that present terms are not sufficient enough to allow *good* expansion decisions. Interfaces should support the identification of relationships between data and suggested query terms.

5 Interactive Video Retrieval

> "As long as one keeps searching, the answers come [Johnson Lewis, 1997]."
>
> JOAN BAEZ
> *American folk singer (born 1941)*

After introducing the main features in video retrieval it is mandatory to have a closer look on the idea of retrieval engines. This chapter gives a survey about interactive video retrieval. Chapter 5.1 explains the concept of video surrogates. In chapter 5.2, the problem of representing videos for browsing using *good* keyframes or fast forward techniques is introduced. In chapter 5.3, a survey is given on video indexing methods. Chapter 5.4 provides an overview on users' video relevance criteria. In chapter 5.5, the most important approaches are presented.

5.1 Video surrogate

Video search engines shall assist users in finding the videos they want. Often, these videos are related to a particular topic which is described using both images and text. This makes it more difficult, as the user needs visual information like keyframes or video playback to judge if a video clip is relevant or not. The text alone is not sufficient enough to find the desired video clip. Previous research has been concentrated on text retrieval, so it is a well-studied process. However, video retrieval as a research field is nearly untouched.

Ding et al. [1999] provided a concise representation of videos called video surrogate. It is also referred to as video abstraction [Lienhart et al., 1997] or video summary [Yeo and Yeung, 1997]. As discussed in [Mu and Marchionini, 2003a], video surrogate can be classified into *visual* surrogate and *textual* surrogate. Textual surrogates contains metadata information such as title, publisher, date, content abstraction, closed-caption data and/or full-text transcript. Textual metadata are useful for textual search.
Video frames or a "skimmed" video of the original [Christel et al., 1999] are referred to as visual

surrogates. The image features are useful for the comparison of keyframes. In some cases, a set of frame images – a filmstrip – represent a video while in other cases, a single keyframe represents its video. Wildemuth et al. [2003a] investigated four variations of one form of video surrogate. They tested different speeds of fast forward by selecting and displaying every *N*th frame from the original video. Based on their user study, they recommend a fast forward default speed of 1:64 of the original video for representing a video. Additionally, users should be able to control the playback speed to adjust to personal preferences.
Hughes et al. [2003] report on an investigation of digital video results pages containing textual and visual surrogates. Participating users were eye-tracked to find out what is more important for users: text or pictures. Their study demonstrated that user statistically reliable concentrate longer on text than on images. Most people use text as an anchor for making a first judgement about a video.

Wildemuth et al. [2003b] compared the effectiveness of a features-only search system, a text-only search system and a system combining both. Their result was that users achieved a higher recall in less time per search with both the transcript-only and the combined system. They also measured the satisfaction of the participants. Also, the transcript-only and the combined system performed better than the features-only system. Their conclusion is that searching for visual features can become a useful supplement to transcript-only searching. A challenge in the video metadata authorisation is how to integrate the visual video metadata with the textual video metadata.

5.1.1 Measuring User Performances

Video surrogates can be classified into five types [Yang et al., 2003] : text surrogates, still image surrogates, moving image surrogates, audio surrogates and a combination of these different types – multimodal surrogates (see table 5.1). *Text surrogates* combines all kinds of bibliographic metadata information. *Still image surrogates* include the video content after extracting the keyframes. *Moving image surrogate* is similar to the original video content as it contains action. *Audio surrogates* represent extracted audio data such as environmental sounds, music or people's dialogues. *Multimodal surrogates* combine audio, visual and textual information.

As these different surrogates have been developed, it is mandatory to develop a method for evaluating the effectiveness of the methods [Goodrum, 2001]. The methods used to evaluate surrogates in textual datasets are inappropriate [Yang et al., 2003] as these measures are also text based and therefore limited in their ability to consider the multimedia characteristics of video surrogates. In [Yang et al., 2003], the researchers propose two general classes of user tasks – recognition tasks and tasks requiring inference – for which they developed performance measures. These two tasks

Type of Surrogate	Examples
text surrogate	title, keyword, description
still image surrogate	poster frame, filmstrip, slide show, video beam, keyframe-based table of contents
moving image surrogate	skim, fast forward
audio surrogate	spoken keywords, environmental sounds, music
multimodal surrogate	text surrogate & still image surrogate, still image surrogate & audio surrogate

Table 5.1: Examples of video surrogates [Yang et al., 2003]

cover the user's ability to remember objects or actions in a video surrogation. The *recognition task* combines object recognition (textual or graphical) and action recognition. The *inference task* combines gist determination (free-text or multiple-choice) and visual gist determination. This categorisation is consistent with the way viewers perceive and understand images [Greisdorf and O'Connor, 2002].

Mu and Marchionini [2003b] introduced four statistical visual feature indexes and suggest to add them to the video surrogate: SLM (shot length mean) – the average length of each shot in a video; SLD (shot length deviation) – the standard deviation of shot length for a video; OND (object number deviation) – the standard deviation of the number of objects per frame over the whole video and ONM (object number mean) – the average number of objects per frame of the video. The features can be used to indicate when a video contains rapid shot changes (*"I am looking for a video that goes fast"*) or slow shot changes (*"old style, leisurely video"*) or when it contains only a few objects in the frame (*"a video that looks simple and clean"*).

5.2 Visual Presentation

Selecting *good* keyframes is an important issue. Empirical studies [Lieberman, 1965] evidence that people have superior memory for images than for text. But in general, details of a picture are not so well remembered [Mandler and Ritchey, 1977]. Ponceleon et al. [1998] argue that *"observers do not remember the scene per se. Rather, they remember the gist of the scene*". Admittedly, there is no *consistent* gist understanding, it rather depends from person to person as people remember different things from the same image. Yang and Marchionini [2005] conducted a study to detect the elements that constitute the *"visual gist"* in the users' mind:

- **Object** such as cars and bridges were the most frequently mentioned elements

- **People** with specific characteristic such as age, gender, dress were the second most mentioned element
- After watching a scene, people got a general impression of the **setting/environment** of the scene
- Users often remembered an **action/activity** or specific **event** they saw
- They remembered about the **theme/topic** such as *Middle East* or *history*
- People identified the **time setting or period** according to objects they saw
- **Geographical location** such as *Egyptian environment* were remembered
- They infer a **plot** to determine whether an object or person was present or not
- **Visual perception** were often mentioned

They concluded that images representing videos should be selected according to the motives they present.
A user study from Ding et al. [1999] conducted that participants more often paid attention to keyframes with one of the following features: text in picture, symbols, novelty, interaction information, emotion and people.

As argued by Lindley [1997], automatically generated visual description *"alone provide very limited effectiveness for applications concerned with what a video stream is "about"'*. There is still the need to add more rich text that contains more information about the semantic meaning of the video part. Especially in scenes where "talking head" holds a lecture. Its visual information are very limited in proportion to its semantic content. Thus, an effective browsing needs a combination of a visual representation and various metadata of the material, as argued by Srinivasan et al. [1997].
Mu and Marchionini [2003a] developed a tool called VAST (Video Annotation and Summarization Tool) for integrating both semantic and visual metadata. Its resulting metadata are a key component for the Open Video Digital Library Toolkit[1].

5.3 Video Indexing

There are two means to authorise video surrogates: by humans or automatically.
According to He et al. [1999], the manual authorisation is more accurate but very time consuming. The automatic metadata authorisation usually utilises videos' physical features such as motion,

[1] see section 5.5.1

shape, colour or brightness.
Images and videos are traditionally indexed manually, a method called *concept-based* video indexing [Enser, 2000]. In this approach, linguistic terms are used to represent, index and retrieve the non-linguistic content. However it can become difficult for the user to use words to describe a multimodal video he has in mind. Thus, a method that combines textual, visual and spatial information is needed to help users in forming their queries. This new approach is called *content-based* video indexing approach. Videos can be indexed based on low-level features such as colour, texture and shape and on high-level features such as events, people or objects.
Concept-based video indexing methods are highly expressive but also, it involves a loss of information during the media transformation process. And of course, it requires more human labour. Content-based indexing methods can be automated and can be cheaper and faster. However, they have the limitation of the semantic gap between the users' queries and the content feature that can be detected and indexed automatically. Yang et al. [2004] and Browne et al. [2002] tested the performance of a concept-only retrieval system and a combined system. No significant difference in performance were detected. Further analysis showed that concept-based video retrieval is working best for *specific* search topics. The combination of concept- and content-based video retrieval showed advantages for *generic* search topics such as *"road with vehicles"*.

According to [Munesawang and Guan, 2005], interactive systems need a *self-adaptation* process to achieve a high retrieval performance under a minimal user input. Traditionally, the relevance feedback paradigm is entirely dependent on the amount of feedback samples [Naphades et al., 2001] and the ability of the searcher to give a consistent feedback.

5.4 Relevance Judging

For creating a retrieval system that supports the user, it is important to find out more about his needs and preferences [Payette and Rieger, 1998]. It is mandatory to find out how people make relevance judgements when searching for video data. Relevance is one of the central concepts in information science. The two most common criteria to evaluate the effectiveness of information retrieval systems – recall and precision – are relevance-based. Two different definitions for relevance exist in literature: *system-oriented* relevance and *user-oriented* relevance [Yang and Marchionini, 2004]. The focus in the system-oriented definition is set on the relations between a specified retrieval request and the returned documents. The user-oriented definition is concentrated on the relations between the users' information needs and the retrieved documents.
Yang and Marchionini [2004] interviewed various experts to find out *"what relevance criteria do people use when they search videos, and in particular, what visual criteria do they apply"*. An

analysis of their interviews generated three categories of relevance criteria: *textual, visual* and *implicit* criteria. The users started their video selection generally based upon textual information. They provided topicality, recency, authorship, genre, duration, reviews or prices. Topicality was the most important criteria for them. Nearly all participants wanted to see some visual information such as videos or images before making a final selection. They mentioned different visual criteria they were interested in such as cinematography, objects/events, motion and style. Sometimes, the final selection about which video to choose was not affected by the actual video content, but by some subjective or implicit criteria such as personal interest, familiarity, accessibility or suggestiveness. The result of this interviews are in line with the results regarding image relevance judgements reported from Markkula and Sormumen [1998].

5.5 Approaches

A wide variety of participants from industry and academy participate in the annual TRECVID workshop.[2] Here, the most successful systems are based on different approaches:
The Dublin City University system supports an image-plus-text search and, for query refinement, a relevance feedback mechanism. The user may decide for each search which features of a video or image similarity shall be taken into account for refinement [Cooke et al., 2004].
The system developed at the Imperial College offers the user the possibility to weight various image features e.g. example-based search, a relevance feedback system and a visualisation system that also presents keyframes that are *close* to a selected keyframe [Heesch et al., 2004].
The system of Amsterdam University (MediaMill) is based on a powerful semantic concept detection system. Users can search by keyword and example as well as by concept [Snoek et al., 2004].
Informedia from Carnegie Mellon University includes the technology for image video feature detection and enables the searcher to weight under these aspects [Christel et al., 2004].

5.5.1 The Open Video Digital Library

The University of North Carolina at Chapel Hill established an open repository of videos which can be used in a variety of ways. Providing this digital library – it is called *Open Video Digital Library*[3] – is motivated by theoretical and practical goals.
A theoretical goal is to evaluate the *Sharium concept* for digital libraries [Marchionini, 1999]. This idea takes the leverage human time, afford and resources into account. Thanks to the Internet, it is

[2] A more detailed survey on TRECVID and its systems will be given in chapter 6.
[3] `http://www.open-video.org`

easily possible to get people involved which is very important, as a digital library has no physical place or reference support like a classical library.
Another theoretical goal is to understand the browsing and searching via electronic equipment. In classical libraries, catalogues and pointer information are positioned more far away from the texts, tapes and other media. A digital library offers both the actual text and index aides in the same interface. For users, this is more convenient. Interestingly, this capacity leads to new behaviour and information-seeking strategies [Marchionini, 1995]. So, the digital library offers the opportunity to study this behaviour.
Thirdly, a goal is to evaluate a framework for digital library interfaces. As they are analogue to the library space and the librarian services, they are most important for the success of digital libraries.
One practical goal of the project is to set up a digital library for research, development and testing. Content characteristics like the visual quality of available videos are relevant for the testing result. An open library has advantages for research groups in different ways: At first, they do not have to worry where they can obtain video data for their research. Then, using the same video data makes the work of different groups comparable as all have to deal with the same quality of data.
A practical side effect is the chance to provide library science students the possibility to test and train their skills on digital library systems.
Finally, an overall idea is to offer an open repository for digital library to the public [Marchionini and Geisler, 2002].
"The OV aims to archive video that people or institutions want to share with the education and research communities" [Marchionini, 2003].

5.5.1.1 Evolution and current Status

First efforts from the University of North Carolina at Chapel Hill to provide a digital library started in 1996. At that time, they worked with Discovery Channel videos with a view to provide material to middle school science in the Baltimore Learning Community Project. Therefore, they indexed short segments of the videos and joined them with images and text and hyperlinks in a dynamic query user interface [Marchionini et al., 1997].
In 1999, the project evolved. The researchers started creating a publicly accessible digital video repository. The usefulness about such an repository was discussed at both the SIGIR workshop and at a retrieval symposium hosted in Chapel Hill. In this year, a first framework was implemented [Slaughter et al., 2000].
The initial framework provided 120 files in MPEG-1 format. They where segmented into eight different programs obtained from the U.S. government. In Spring 2000, videos collected for the Carnegie Mellon Informedia Project, Prelinger Archives, and the Howard Hughes Medical Institute

were added to the repository [Marchionini and Geisler, 2002].
In 2004, they collected 2039 video files, all together half a terabyte of size. Table 5.2 (taken from [Geisler, 2004]) shows the structure at that time:

Genre	Duration	Colour	Sound	Format
Documentary: 494 Educational: 186 Ephemeral: 1140 Historical: 187 Lecture: 16 other: 5	$<$ 1 min.: 182 1 to 2 min.: 249 2 to 5 min.: 340 5 to 10 min.: 320 $>$ 10 min.: 918	Colour: 988 B&W: 1040	with sound: 1643 silent: 385	MPEG-1: 1403 MPEG-2: 1067 MPEG-4: 409 Quicktime

Table 5.2: Characteristics of 2004 OVDL content

According to the Interaction Design Laboratory [2006], the repository currently contains eight different collections:

1. University of Maryland HCIL Open House Video Reports
2. The Informedia Project at Carnegie Mellon University
3. Internet Moving Images Archive
4. 2001 TREC Video Retrieval Test Collection
5. CHI Video Retrospective
6. Digital Himalaya Project
7. NASA K-16 Science Education Programs
8. William R. Ferris Collection

The developers at Chapel Hill focus their work on user interface development. That is why they aim to use as many open source products to set up and run the system. Digitalisation of the available video data is done in their Interaction Design Lab. However, newer files are already submitted in digital form. The segmentation is done manually by students. It is considered as a good exercise for them to get in touch with the video material. For keyframe extraction, the staff mostly used the MERIT software suite from the University of Maryland [Kobla et al., 1998]. The keyword identification also is mainly done manually [Marchionini and Geisler, 2002].

5.5.1.2 Graphical Interface

The Open Video Project provides a web-based user interface which was redesigned in 2004. Mainly, the visual style was redeveloped, as now, CSS files are created for layout and style pages. The optimised CSS are the best possible compromise between functionality and appearance in supported browsers [Geisler, 2004].

Figure 5.1: Open Video Graphical User Interface – Start Page [Geisler, 2004]

Figure 5.1 shows a screenshot of the actual interface. It provides the chance to trigger a quick search in entering a search query and the chance to browse through the collections according to different features (compare to table 5.2). On the front page, it also lists all collections so that the user can quickly access them. A special feature can be found on the right-hand side of the interface: The newest and the most popular videos are listed separately to catch the user's eye.

The search results (see figure 5.2) are listed in a classical way starting with the most relevant results. The order and the size of the result display can be changed manually by the user. This visual preview presents a sample keyframe and the most important metadata concerning each result. On the left-hand side of the result list, the user also gets the opportunity to modify his search criteria

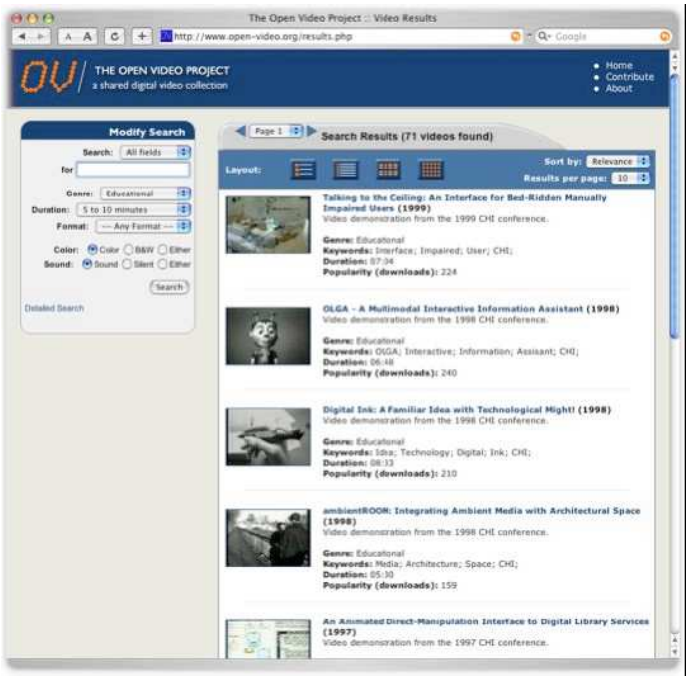

Figure 5.2: Open Video Graphical User Interface – Result Listing [Geisler, 2004]

5.5.2 YouTube

Interactive web-based Video Retrieval is getting more important as both retrieval giants Yahoo![4] and Google[5] are working on their own video retrieval engine. In addition there are numerous video search engines such as `www.truveo.com` and `www.blinkx.com`. However, another system called YouTube[6] – firstly presented in February 2005 [Jefferson, 2005] – is currently more popular than its competitors. Alexa Internet ranks `YouTube.com` as 20^{th} most visited site on the net [Alexa Internet, 2006].

YouTube is a website that allows users to upload and share videos. The uploaded video collection consists of movies, TV show clips, music videos and home videos. A retrieval is based on text queries indexed using a concept-based method. To play videos, YouTube uses the Macromedia Flash technology. These video feeds can easily be embedded into Weblogs or other websites like MySpace [YouTube, 2006b].

[4] `http://video.search.yahoo.com`
[5] `http://video.google.com`
[6] `http://www.youtube.com`

5.5.2.1 Concept

In March 2006, approx. 20.000 new videos were uploaded every day shared with millions of users [Woolley, 2006]. The problem with this content is that most of it is not of a good quality. As digital cameras and simple publishing software is relatively cheap, it is natural that there is such an explosion of media creation. Although the platform will have an important place as social phenomenon – “hey guys, check out this cool video!” [Weber, 2006].
When uploading a video, the user manually has to describe the video and classify it into the following categories [YouTube, 2006a] using keywords:

1. Art & Animation
2. Autos & Vehicles
3. Comedy
4. Entertainment
5. Music
6. News & Blogs
7. People
8. Pets & Animals
9. Science & Technology
10. Sports
11. Travel & Places
12. Video Games

These keywords and the descriptive text is used for retrieval.
It is not allowed to upload copyright protected video material, but due to the mass of uploads and the miss of control, such material can be found continuously. In general, YouTube only discovers these files when they are reported by users or the original copyright holder. So in February 2006 for instance, YouTube was forced to remove copyrighted NBC video clips from their site [Woolley, 2006].

5.5.2.2 Graphical Interface

According to Shannon [2006], YouTube's main interface is the main reason for the success of this start-up company. It has, except for minor changes, remained structurally unchanged since its start in 2005. As many videos are updated every day, it is guaranteed that the content of the page changes continuously. So it has the potential to stay interesting for visitors.
Although, one condition to stay interesting is the need to organise the content. The interface makes it easy to search and filter what users want to see. Important is also the relationship between user and publisher. Consumers join in the responsibility of publishing while publishers focus on offering the best platform. It makes it easy for users not only to share their videos but also commenting other users' video publications.

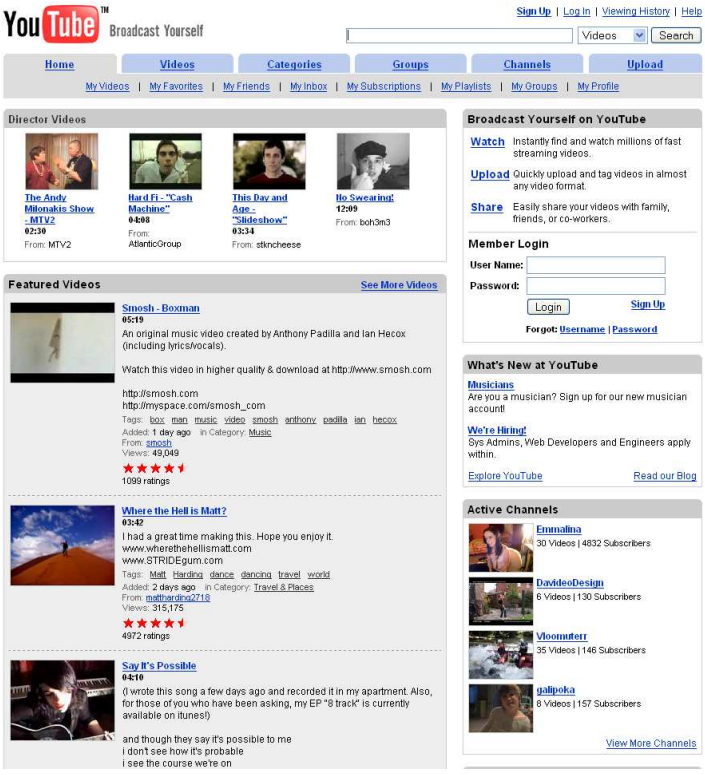

Figure 5.3: YouTube Graphical User Interface

Figure 5.3 shows the web interface of YouTube. On the top of the start page, the user can enter a search query. The page itself has a very simple structure. The user can navigate using tabs. On the *Home* tab, the most popular videos are listed in the centre of the page. For each video entry are specific information available: A title, the duration of the video, a textual description, keyword tags, the date when it was added to the collection, the owner, how often it has been seen and a rating (symbolised with stars). The *Video* tab shows alternatively the most viewed, top rated, most

discussed, most linked, recently featured, most recent, the top favourites or random videos.
Clicking the *Categories* tab, the user can browse through the defined categories.
The *Groups* and *Channels* tab combines specific keywords to groups or channels of interest, e.g. the tags “Football Soccer World Cup” form the group “World Cup”.
Clicking on the *Upload* tab, the user can log in and then upload and classify his videos.

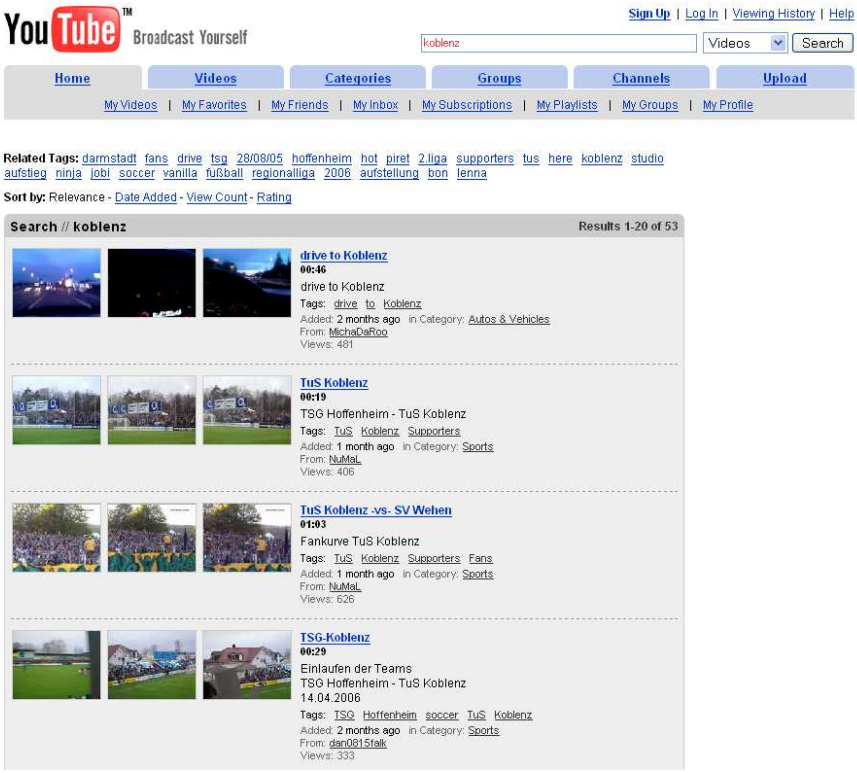

Figure 5.4: YouTube Graphical User Interface – Result Listing

Figure 5.4 shows the result listing of a retrieval. On the top, it lists the most related keywords which have been determined according to other keywords that users applied besides the search query. Clicking on them, a new retrieval starts using this keyword. Every page lists the maximum of 20 results. Each result is presented showing the available textual information (see description above) and three random keyframes.
When clicking on a result, the video can alternatively be played in full screen modus or in a small window of the browser using the Macromedia Flash technology. It is stored in Flash Video format (.flv). While playing, the video is downloaded into the cache. When the user is logged in, he can here rate the video and also comment it.

6 TRECVID

> "The only source of knowledge is experience [Mayer and Holms, 1996]."
>
> ALBERT EINSTEIN
> *German physicist (1879 – 1955)*

To reach comparable research in any scientific sector, it is necessary to provide scientists a broader platform for timed presentation and alteration of their work, especially in an advanced and continuously changing sector. One of these platforms is the TRECVID workshop which is presented in chapter 6.1. The organisers offer a data set to all participants which is described in chapter 6.2. In chapter 6.3, some systems developed by participants of past TRECVID workshops are presented to give an insight and overview of recent developments. To provide comparison between the efficiency of these systems, TRECVID creates search tasks. These search tasks are described in chapter 6.4.

6.1 Text REtrieval Conference (TREC) and TRECVID

The Text REtrieval Conference (TREC), co-sponsored by the U.S. Department of Defence and the National Institute of Standards and Technology (NIST) supports research of information retrieval groups by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. Its goals are:

- to encourage research in information retrieval based on large test collections
- to increase communication among industry, academia, and government by creating an open forum for the exchange of research ideas
- to speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems; and

- to increase the availability of appropriate evaluation techniques for use by industry and academia including development of new evaluation techniques more applicable to current systems.

NIST provides a test set of documents and tasks for each TREC (read more about it in the sections 6.2 and 6.4). The participants develop their own retrieval systems using these data sets and send a list of their top-ranked documents to NIST (the most effective systems are introduced in chapter 6.3). The institution then judges the documents for correctness and evaluates the results. Every TREC circle ends with a workshop [NIST, 2004b].

In 2001, the TREC workshop was opened for the Video Track. Its goal was to push progress in content-based retrieval from digital data. The test set builds on available video files provided by the Open Video Project of the University of North Carolina at Chapel Hill, the NIST Digital Video Library and stock shot video provided by the British Broadcasting Corporation. It consisted of 11 hours of videos in the MPEG-1 format.
This TREC Video Track had 12 participating groups, 5 from Europe, 2 from Asia and 5 from the United States [Smeaton et al., 2002].
Beginning in 2003, the track became an independent evaluation called TRECVID. It is coordinated by Alan Smeaton (Dublin City University) and Wessel Kraaij (TNO Information and Communication Technology). Paul Over and Tzveta Ianeva provide support at NIST.
All participants got a copy of approx. 120 hours (241 30-minute programmes) of ABC World News Tonight and CNN Headline News recorded by the Linguistic Data Consortium from late January through June 1998. Moreover, approx. 13 hours of C-SPAN programming (approx. 30 mostly 10- or 20-minute programs) about two thirds from 2001, others from 1999, one or two from 1998 and 2000. The C-SPAN programming includes various government committee meetings, discussions of public affairs, some lectures, news conferences, forums of various sorts, public hearings, etc. [NIST, 2004a].

In February 2006 till November 2006, the NIST calls for participation in the 2006 TREC Video Retrieval Evaluation. Participating groups have to test their systems on one or all of the following four tasks/evaluations and share their results [NIST, 2006a].

- shot boundary detection
- rushes exploitation
- high-level feature extraction
- search (interactive, manually-assisted, fully automatic)

In 2006, Glasgow University will participate for the first time in TRECVID. Its chosen tasks are rushes exploitation and the search task [NIST, 2006d]. Dr. Xavier Hilaire and Jana Urban are working on the fully automatic search while the interactive search is topic of this thesis. Thereinafter, an article will be written for the workshop which describes the work and the gained results.

6.2 2005 Data Set

Due to agreements with the providers of the data and the supporting associations, the TRECVID data is not available for everyone. TRECVID participants received the main data on a hard disk after signing a contract. The set is provided by NIST and the Linguistic Data Consortium [NIST, 2006e]. A part of it is publicly available and can be downloaded from the web. [1]

According to NIST [2006e], the 2005 collection contains:

- broadcast news video files in MPEG-1 format
- master keyframes
- shot boundary annotation
- low-level feature truth judgements
- high-level feature truth judgements
- search relevance judgements
- camera motion annotation (donated by Joanneum and KDDI)
- common development feature annotation (using the CMU and IBM tools)
- low-level development features (donated by CMU)
- master shot boundary reference
- search topics and included images
- ASR/MT output
- evaluated system submissions

In the following sub chapters, the most relevant data for this work are presented.

[1] `http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html`

6.2.1 Broadcast news video files

The Linguistic Data Consortium collected video material and provided it to TRECVID 2005. The data is rights secured for research. It contains approx. 170 hours of television news from November 2004.

Language	Episodes	Source	Program	Total (hours)
Arabic	15	LBC	LBC NAHAR	13.13
Arabic	25	LBC	LBC NEWS	23.14
Arabic	17	LBC	LBC NEWS2	6.80
Chinese	28	CCTV4	DAILY NEWS	25.80
Chinese	21	CCTV4	NEWS3	9.30
Chinese	21	NTDTV	NTD NEWS12	9.28
Chinese	18	NTDTV	NTD NEWS19	7.93
English	26	CNN	AARON BROWN	22.80
English	17	CNN	LIVE FROM	7.58
English	27	NBC	NBC PHILA23	11.83
English	19	NBC	NIGHTLY NEWS	8.47
English	25	MSNBC	MSNBC NEWS11	11.10
English	28	MSNBC	MSNBC NEWS13	12.42

Table 6.1: TRECVID 2005 video data

NASA and the Open Video Project provided several hours of NASA's Connect and/or Destination Tomorrow series which have not yet been made public.
The BBC provided about 50 hours of *rushes* on vacation spots. *Rushes* are pre-production travel video material with natural sound and errors.
The video data can be used to experiment and to demonstrate functionality which is useful in managing and mining such material.

The video data mainly consists of broadcast news. The 2005 collection is the first collection that also contains sources in Arabic and Chinese language. This matter complicates the search and feature detection tasks, as they introduce a greater variety of production styles. And of course, the text-to-speech contains more errors as an additional fully automatic translation from Arabic and Chinese sources to English has to be done [Over et al., 2005, NIST, 2006e].

The data set is split into two sets: The test data and the development data. A random sample of approx. 6 hours of the television broadcast is combined with about 3 hours of NASA videos as shot boundary test data. The remaining 160 hours of television video data were split in half chronologically by source. One half was used as development data for the search, high/low-level feature

and shot boundary detection tasks. The other half was combined as test data for the search and high/low-level feature tasks. The BBC rush video set was split and designated both as development and test data [NIST, 2006e].
All video data are in MPEG-1 format.

6.2.2 Shot boundary annotation and master keyframes

The master shot reference was provided by Christian Petersohn from the Fraunhofer Institute for Telecommunications in Berlin. Their system detects and determines the position of dissolves, wipes, fades and hard cuts to create a reference. Figure 6.1 gives an overview of their system. A detailed description can be found in [Petersohn, 2004].

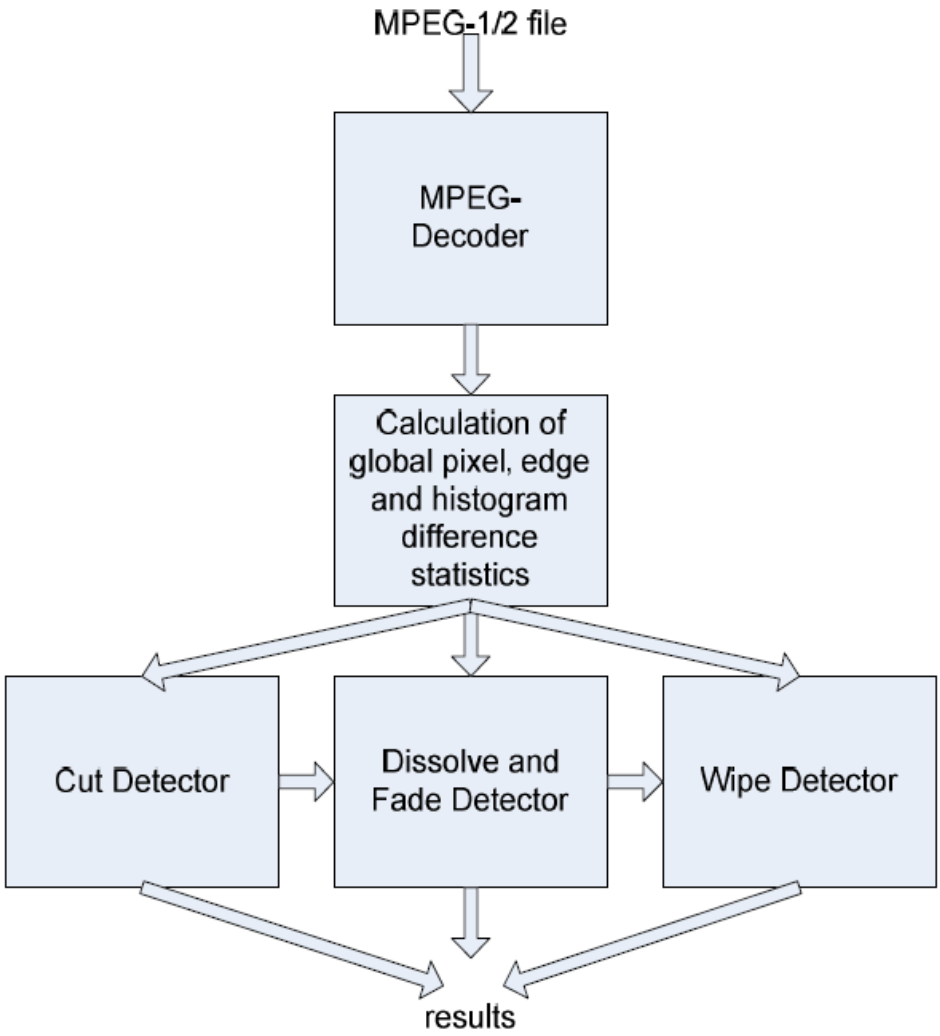

Figure 6.1: Shot boundary detection system: System overview [Petersohn, 2004]

Every video was segmented to create the master list of shots. They call the result of this pass a subshot. The system accepts only shots with a duration of at least 2 seconds in length as master shots. In a second pass, the subshots where aggregated until the current shot was at least 2 seconds in duration. These detected master shots are used in submitting results for the feature and search task [NIST, 2006e].

Their resulting reference is formatted in MPEG-7 and can be downloaded from the Internet. [2] Table 6.2 shows an extract of one file.

```
<MultimediaContent xsi:type="VideoType">
  <Video id="TRECVID2005_141">
    <MediaLocator>
      <MediaUri>20041030_133100_MSNBC_MSNBCNEWS13_ENG.mpg</MediaUri>
    </MediaLocator>
  <MediaTime>
    <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
    <MediaDuration>PT00H27M19S29150N30000F</MediaDuration>
  </MediaTime>
 <TemporalDecomposition gap="false" overlap="false">
   <VideoSegment id="shot141_1">
     <MediaTime>
       <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
       <MediaDuration>PT00H00M02S15075N30000F</MediaDuration>
     </MediaTime>
     <TemporalDecomposition>
       <VideoSegment id="shot141_1_RKF">
         <MediaTime>
           <MediaTimePoint>T00:00:01:7037F30000</MediaTimePoint>
         </MediaTime>
       </VideoSegment>
     </TemporalDecomposition>
   </VideoSegment>
```

Table 6.2: Common shot boundary reference example listing

Dublin City University formatted this reference and created a common set of keyframes. The keyframes were selected by going to the middle frame of the shot boundary and parsing left and right of that frame. The nearest I-Frame became the keyframe. Two different kind of keyframes are provided: on the subshot (NRKF) and the master shot (RKF) level [NIST, 2006e].

[2] http://www-nlpir.nist.gov/projects/tv2005/tv5.master.shot.ref.mpeg7.zip

6.2.3 ASR/MT output

For the 2005 workshop, TRECVID dived into all the complications of cross-language information retrieval. As the videos are in Arabic, Chinese and English language, both automatic speech recognition (ASR) as well as machine translation (MT) play an important role.
ASR systems can be used to transform spoken words into computer text. The system is able to recognise a limited vocabulary. The ASR data for the English and Chinese data set are provided by Alexander Hauptmann from CMU. It was the standard output of a Microsoft Research beta system [Over, 2005], so it had no specific tuning to the data set.
According to Hauptmann [2005], each English and Chinese video has four associated files that are created by the Microsoft Research system.

- *.spchtim* files list the words with start time in 10 millisecond increments
- *.spchtim2* files list the words with start time and end time in 10 millisecond increments
- *.phrase* files list the words with start time and end time in 10 millisecond increments
- *.msasr* files are the direct output of the speech recogniser with time/duration and also contain confidence

Besides these data, NIST provides XML files for every Arabic, Chinese and English video file. It contains some meta data about the video itself and the result of the speech recognition. Figure 6.3 shows an extract of an example file listing the meta data. Figure 6.4 shows an extract of the speech recognition in the file.

```
<video_label>
  <label>
    <field name="Broadcaster" type="string">MSNBC</field>
    <field name="Start_Time" type="int">1114718160</field>
    <field name="Program" type="string">MSNBC News</field>
    <field name="Broadcasting␣Country" type="string">United States</field>
    <field name="Completed" type="string">True</field>
    <field name="Date" type="date">2005-03-01</field>
    <field name="End_Time" type="int">1114720140</field>
    <field name="CC3_Event_ID" type="int">1099511653985</field>
    <field name="Source" type="string">Tape</field>
    <field name="Broadcasting␣Language" type="string">US English</field>
    <field name="Protect" type="string">Yes</field>
    <prop_list />
  </label>
</video_label>
```

Table 6.3: Meta Data example listing

```
<track_list>
  <text_track id="0x574f5244" name="Words">
    <text record_id="1">
      <timespan in_msec="970" in_smpte="00:00:00:29" out_msec="1089"
       out_smpte="00:00:01:03" />
      <prop_list />
      New
    </text>
    <text record_id="2">
      <timespan in_msec="1090" in_smpte="00:00:01:03" out_msec="1530"
       out_smpte="00:00:01:16" />
      <prop_list />
      world
    </text>
    ...
  </text_track>
</track_list>
```

Table 6.4: ASR example listing

Machine Translation (MT) is the automatic translation of text into another language. For TRECVID 2005, it means the automatic translation of Chinese and Arabic texts into English. The machine translation data also was the output of an off-the-shelf product [NIST, 2006e].
The provided file is a translation of the .phrase files that are associated to the Chinese videos. Figure 6.5 shows an example file. The format has to be read in the following way:
start time (in 10 milliseconds) <tab> end time (in 10 milliseconds) <tab> phrase.

```
< REPORT file = 20041101 _ 110000 _CCTV4_NEWS3_CHN >
54523 55035 25 about transferred also to reduce car exhaust gas among the
55193 55985 civilization thus and i certainly .
56055 56588 sina .
```

Table 6.5: Machine Translation example listing

The test data cannot be used for system development. This is what the development data was intended for. Glasgow University uses the 2005 development data set for its research.

6.3 Examples of Video Retrieval Systems

It always makes sense to learn from the Best! In the case of Video Retrieval at TRECVID, it is worth taking a closer look at its most effective systems. These are the *Informedia System* from

Carnegie Mellon University (CMU), *Físchlár Digital Video System* from Dublin City University (DCU) and a retrieval system developed at the Imperial College London (ICL). Concentrating on the tasks shot boundary detection and searching, those systems will be described in the next sections.

6.3.1 Informedia Digital Video Library (CMU)

In 1994, Carnegie Mellon University, Pittsburgh, USA started a project called *Informedia-I: Integrated Speech, Image and Language Understanding for Creation and Exploration of Digital Video Libraries*. Its goal was to develop and to research into technologies for data storage, search and retrieval and for embedding these technologies into a video library system. Their first approach used combined speech, language and image understanding technology to transcribe and index video data [Carnegie Mellon University, 1998]. Adjacent projects made it possible to continue and to enlarge research in the idea of developing a video library system [Carnegie Mellon University, 2006]. Participating in TREC 2001 Video Track and the succeeding workshops, the system had been evaluated in diverse tasks. In 2004, the researchers participated in the semantic feature extraction task and the manual, interactive and automatic search task. For the interactive search, a complete video retrieval system using visual *and* textual data versus a visual-only system has been contrasted. Additionally, they compared expert and lay knowledge users [Hauptmann et al., 2004]. In 2005, they evaluated the system in low-level feature extraction, semantic concept feature extraction task, the search task and the BBC stock footage challenge [Hauptmann et al., 2005].

6.3.1.1 Graphical User Interface

Alexander Hauptmann, Senior Systems Scientist at CMU acknowledged, that the new aspects of their TRECVID 2005 system have not been published yet. Therefore, figure 6.2 illustrates the 2004 interface of the Informedia system which was, according to chief interface architect Mike Christel, nearly identical to the one of the 2005 workshop.

The figure shows the features of the system: The interface consists of different windows. On the top of the interface is a search query text box. Next to it, the topic of the search task and the available time is displayed. Here, the user can also decide to start the next task. On the right-hand side in the so-called answer area, the answers of the test are displayed. These are shots which are declared to be relevant shots by the user.

After entering the query, the system displays the retrieved keyframes in a new window. This window shows thumbnail images representing the video shots. Clicking on one keyframe, the user

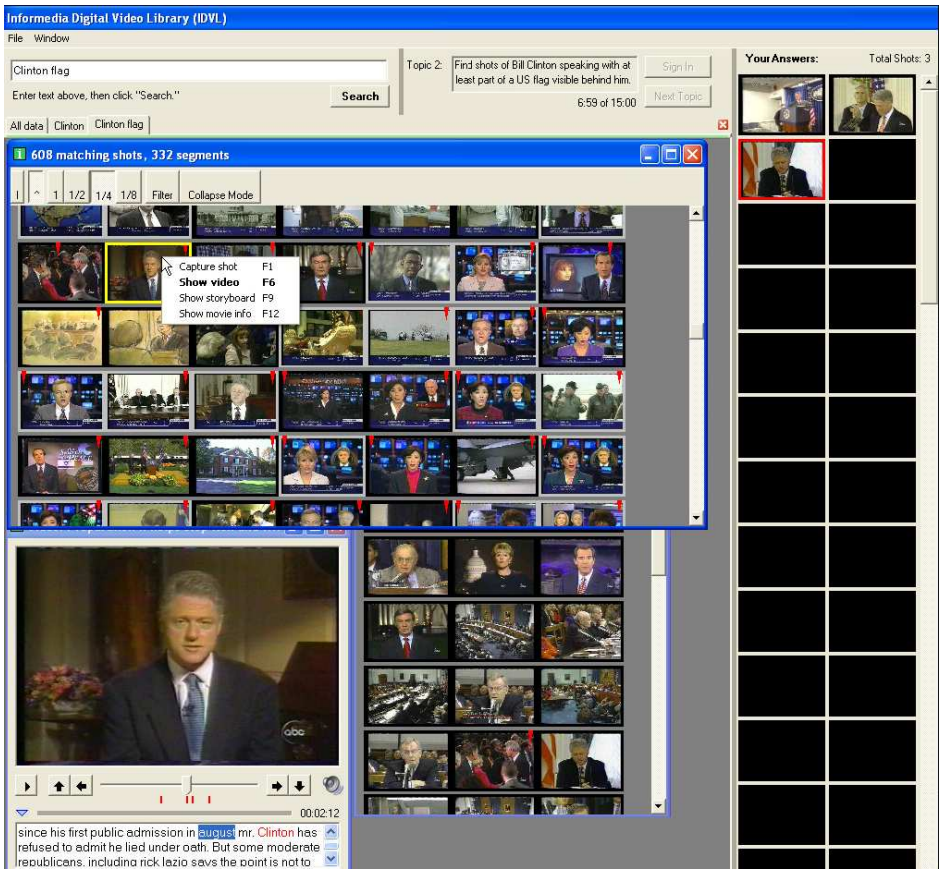

Figure 6.2: Informedia search interface [Christel and Concescu, 2005]

can choose rather to *capture the shot*, *show the video*, *show a storyboard* or to *show further movie information*.

Capturing the shot adds the keyframe to the answer list on the right side. Clicking on *show the video* opens a new window playing the selected video. Spoken text is displayed, highlighted and scrolls while the video is playing. *Show movie info* displays further information concerning the video like title, date of broadcast and duration (can *not* be seen on figure 6.2). Figure 6.3 illustrates the *show storyboard* feature: It lists keyframes arranged in chronological order. The user can inspect the keyframes, play the videos, see other shots in the same broadcast element, receive more information of the news broadcast and capture relevant shots [Christel and Concescu, 2005].

6.3.2 Físchlár Digital Video System (DCU)

The Centre for Digital Video Processing at Dublin City University, Ireland is one of the most important players in the field of video processing. They participated on all video retrieval workshops, besides, this institution is the main coordinator of TRECVID. Their main interest is to develop

Figure 6.3: Storyboard showing “best road shots set” [Christel and Concescu, 2005]

techniques to provide content-based navigation through digital video collections. This includes searching, browsing, filtering, playback, summarising and linking video information. Their flagship is the Físchlár system, which runs with more than 2000 registered users on DCU campus and is available in three different versions:

- *Físchlár-TV*: This system records broadcast TV from any of eight terrestrial broadcast stations
- *Físchlár-News*: It records the daily evening news from the national broadcaster’s main TV channel (RTE1) and automatically segments news story boundaries. So it provides story-based news searching for the users. A variant is *mFíschlár-News* which provides access to a news archive using a mobile device.
- *Físchlár-Nursing*: It conducts access to educational videos for the School of Nursing at DCU.

It is planned to provide access for all university libraries in Ireland [Smeaton, 2002, Centre for Digital Video Processing, 2005].
In TRECVID 2004, DCU researched in the interactive search task. They developed and compared two video search systems based on their Físchlár Digital Video System: one with text and image-based searching, the other one with image searching only [Cooke et al., 2004]. In 2005, they experimented in the automatic and interactive search tasks and the BBC rushes task. They developed a multi-user system using a DiamondTouch tabletop device [Foley et al., 2005].

6.3.2.1 Web Interface

The interface of the Físchlár system can be accessed by using a web browser, either Internet Explorer or Netscape, with an ORACLE plug-in for video streaming [Smeaton, 2002].

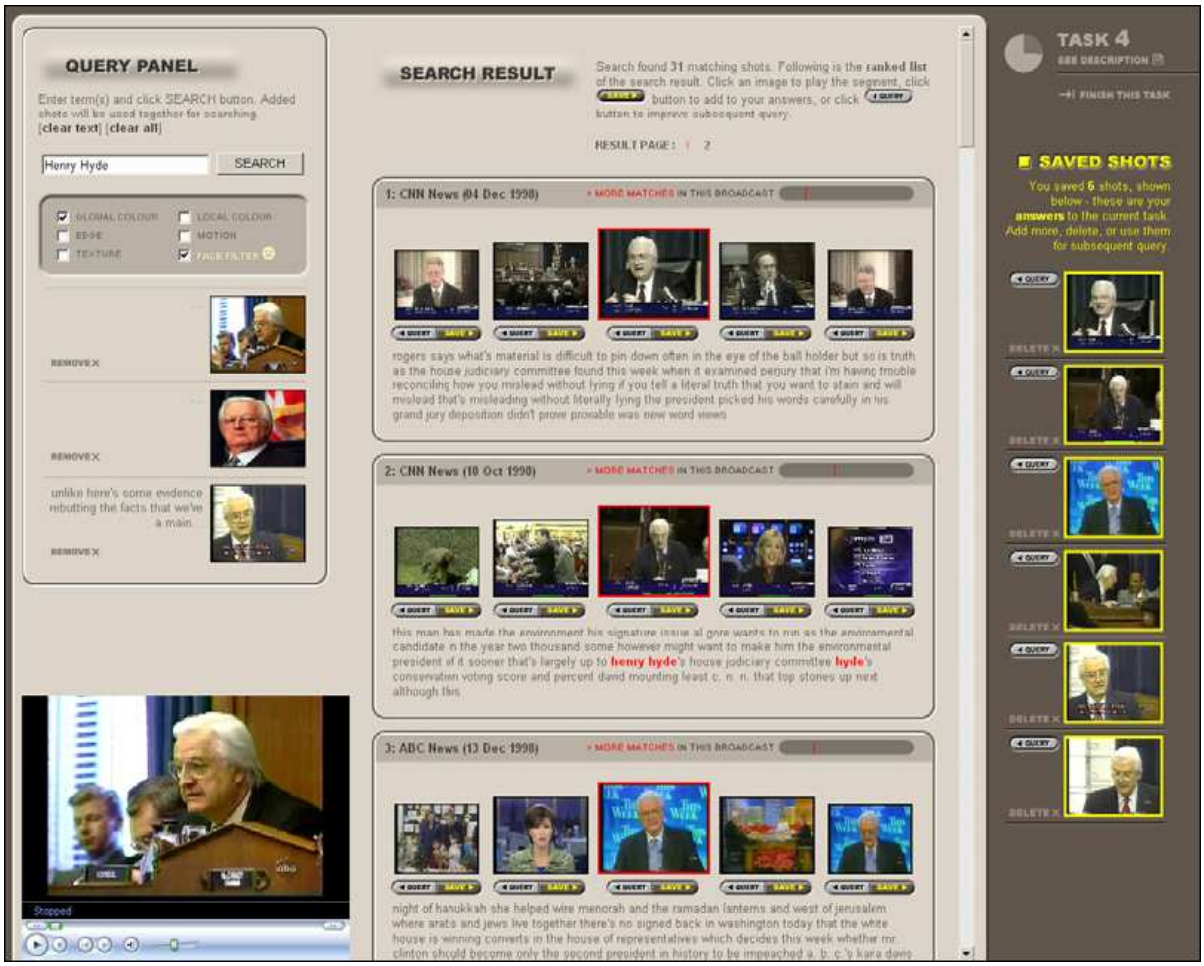

Figure 6.4: Físchlár Digital Video System [Cooke et al., 2004]

They designed it after testing diverse screen mock-ups in the very beginning of system development, discussions about them and phases of iterative refinement. Every year, they tried to rectify the interface in considering, what was good about it and what were the problem elements in the last year's user experiment.

In this thesis, it is not expedient to compare their 2005 DiamondTouch tabletop approach, as it differs a lot to the other systems.
Therefore, figure 6.4 shows a screenshot of the 2004 workshop implementation. That year, they divided their interface into a Administrative Area and a Working Area. The Administrative Area is

placed on the right-hand side and contains information useful for the specific search task (searching for videos for a limited period of time). This includes the task number, a clock showing the remaining time, the task description, and a list of the relevant shots found by the user. The Working Area is split into a query panel and a search panel.

Using the text and image-based system, the user can enter a query and/or add an image. Additionally, he has to check at least one of the six checkboxes: *Global Colour, Edge, Texture, Local Colour, Motion, Face Filter*. They correspond to the six different visual features that are used for the visual retrieval process. After selecting, the user has to press the *search* button to trigger retrieval. The result will be presented in the search result area, which is situated in the middle of the screen.
Every match is surrounded by two preceding and following shots respectively to provide the context of the result. The matched keyframe, which might be the most important, is displayed largest and has a red box surrounding it. Neighbours on both sides are smaller. Each result displays some textual information like the date and the name of the broadcast and contains also a timeline presenting the approximate position of that keyframe in the whole video. Físchlár provides a mechanism to browse through the entire broadcast from which the keyframe was found. Using this function, the different results are marked at the timeline, which allows the user to jump immediately to the relevant frame.
According to this, the interface offers three different ways of browsing: initial search result, more matches within one broadcast and a full broadcast.

Under every keyframe, there are two buttons for supporting relevance feedback. The *Add to Query* button adds the accordant keyframe to the Query Panel. After pressing the *search* button again, the added frame will be part of the new query. The second button, the *save* button is for saving the shot to the Administrative Area [Cooke et al., 2004]. The most important part is the playback feature. Físchlár uses the Microsoft Media Player to play selected videos.

6.3.3 iBase (ICL)

The Imperial College London participates in many projects concerning multimedia and information systems [Imperial College London, 2006]. Thus, they have much experience in the field of video retrieval. Like CMU and DCU, they have contributed research to the TRECVID workshops and its predecessors. They have been developing their system for a long time now, however the name of it has changed continuously.
In TRECVID 2004 and 2005, they experimented in shot boundary detection, high-level feature extraction, search and story boundary detection tasks. In shot boundary detection, they used a

colour-histogram detection method. The search task is complemented with relevance feedback [Heesch et al., 2004, Jesus et al., 2005].

6.3.3.1 Graphical User Interface

The interface of the ICL system unites text-based search, content-based search with relevance feedback and temporal browsing into a unified interface. Although they included many different techniques in their system, they have an emphasis on user interaction and user navigation.

Figure 6.5 shows a screenshot of the interface as shown at the 2005 workshop. The search process is divided into two phases. In the first phase, the user can enter any search query on the left-hand panel. By default, the system uses textual queries, but the user can modify it and include visual search using a pop-up. After triggering the retrieval, results are listed line-by-line in the centred panel of the interface. The most relevant image is listed in the top-left corner while the least relevant can be found in the bottom-right corner of that panel. The results are divided into different pages to avoid too many images on one page. A selected image has a red border surrounding it. At the bottom of the interface, this image is displayed in its context. This feature called *temporal browsing* shows the temporal neighbours of the selected image using a fisheye visualisation. These neighboured images get smaller, the bigger the distance to the main image is.
The interface offers the user 4 different low-level texture and colour methods for searching as well as a textual search: *Tamura Features*, *Gabor Filter*, *RGB colour histogram*, *marginal HSV colour moments* and *bag-for-words feature*.
Still in the first search phase, the user can change his query and/or add images to it. In the second phase, the system asks the user to classify the results he got [Jesus et al., 2005].

6.3.4 Summary and Discussion

The interfaces mentioned above have some differences, but also some common features. This section compares these differences and discusses, which approaches are more useful or more effective. The interfaces are structured having three elements: A *search panel*, a *result panel* and a *playback panel*. All these elements are absolutely necessary. The search panel is for building a query (textual and/or visual), the result panel is needed for showing the retrieved keyframes. The playback panel is necessary for playing and stopping selected video segments.

Figure 6.5: Imperial College London System [Jesus et al., 2005]

6.3.4.1 Search Panel

All interfaces support textual and visual-based search. As mentioned in [Mezaris et al., 2004], a visual similarity re-search using a sample picked keyframe is a suitable feature in retrieval.
They have the main feature of the interface, the search panel, placed on the left side. Physiological studies [Maass and Russo, 2003] revealed that people reading from left to right tend to position a subject to the left of an object first. As the authors of the interfaces are mainly socialised in Europe and America, it is consequent for them to place it in that position. The images for visual-based search are put close to the query text box. This is due to the Locality Principle, as it is propagated in Software Engineering. An interesting article about this principle is [Denning, 2005]. The human short-term-memory has only a capacity of approximately 7 seconds [Ingber, 1985]. Therefore, it is also expedient to show these images permanently, as the user maybe wants to compare them with new retrieved keyframes.
Físchlár and the ICL system use different search techniques and give the user the opportunity to

choose between these techniques. This makes it more confusing for novice users as they do not know the differences for sure. Anyway, this feature is imperative as these diverse techniques give different efficient results.

Físchlár as well as the ICL system has a button for ending the search and starting a new search. This is necessary because of the relevance feedback. If they would not provide this button, the system could not recognise the start of a new independent search and use former results for relevance feedback. In all likelihood, it would bring useless results. Informedia does not need this feature as it does not support relevance feedback techniques.

6.3.4.2 Result Panel

Bigger differences can be found in the presentation of the results. All systems use keyframes to show the results. The ICL system and Informedia list only the relevant images in an order common for search engines. This more traditional listing is very simple and uses the maximum place available. Though, it is difficult, if not even impossible, to find out the context of the displayed keyframe. Another approach is implemented in the DCU system: Their interface shows every result in its context by showing neighboured frames in a fisheye visualisation. This is useful, as a user often can find more relevant images in the neighbouring frames. The ICL system picks up this approach in their temporal browsing feature. For shot level content-based retrieval, it is the most common navigation method [Heesch et al., 2004, Wildemuth et al., 2003b].
The characterisation of the results using textual information like the name and the date of the broadcast or the approximate location of this shot within the broadcast should not be underestimated. Furthermore, text extracted from Speech Recognition Software gives more information about the context of the result. Both features are implemented in Físchlár, but not in the ICL system. Informedia supports this approach, but it is necessary to do an extra click to receive such information.

6.3.4.3 Playback Panel

A very important feature is the playback of the video. The systems use the Microsoft Media Player, which makes them operating system dependent. But as the main focus of all systems is in video retrieval and not in developing an operating system independent tool, their solution is acceptable.
Different from the others, Informedia highlights the words from text recognition software while they are spoken in the video. This is a nice feature especially for non-native speakers, but not compulsory.

According to this analysis and after internal discussions in the IR Group, some proposals for a possible interface for the system have been worked out. They are illustrated in appendix B.

6.4 TREC Search Task

As mentioned in chapter 6.1, groups participating in TRECVID may test and train their systems by using a huge video data set in MPEG-1 format. (The 2006 video data for instance consists of more than 210 hours of television news from November 2004 in English, Arabic and Chinese language [NIST, 2006c]). However, one condition to use this data set is that every team has to test their system using query-based and browsing search tasks alleged and defined by NIST. Dependent on multiple relevant shots they found coming from more than one video, NIST personnel viewed the videos (with sounds turned off) and created different topics for creating search tasks: generic/specific and person/thing/event [Smeaton et al., 2004].
After analysing the TRECVID 2004 and 2005 video collection, NIST recommended the 2 * 24 tasks listed in Appendix A.

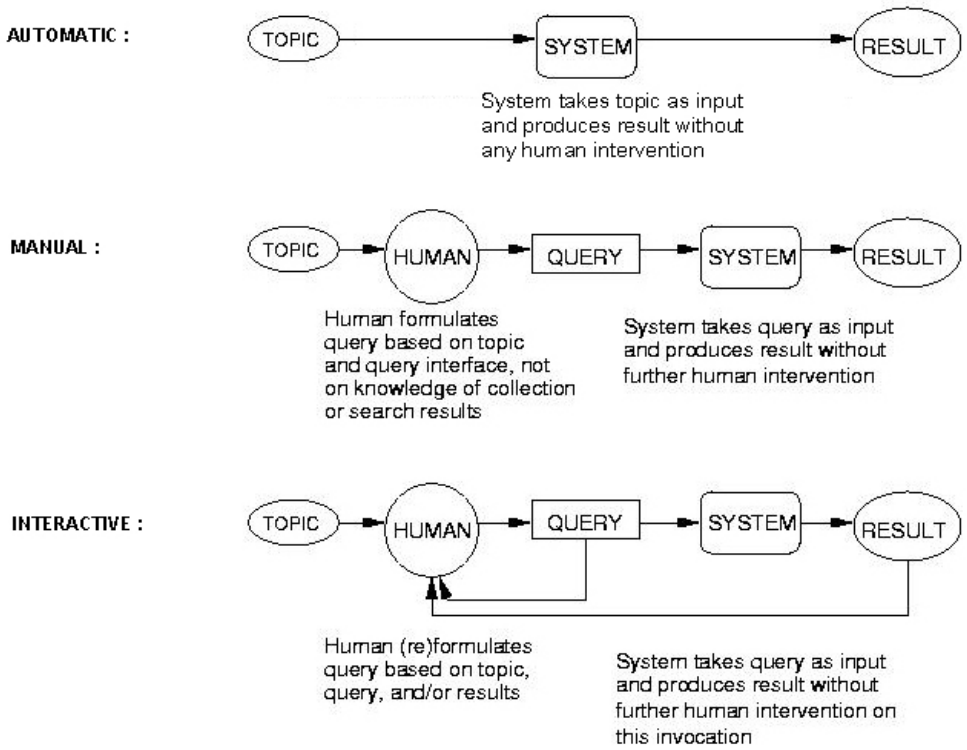

Figure 6.6: TREC Search Tasks [NIST, 2006b]

In TRECVID 2005, as illustrated in figure 6.6, *fully automatic search submissions* (no human input into the loop) as well as *manual* and *interactive submissions* were accepted [NIST, 2006b] while in TRECVID 2004 the fully automatic analysis and query generation was set aside [NIST, 2005]. Inevitable was that one baseline run was required for the manual and the automatic system. Furthermore, all manual runs within one site had to be carried out by the same person to enable comparisons between all participating groups. Hence, the searcher should switch between the different search variants. Important for a comparable test result was that the searchers were no experts in a given topic but had a good educational background. This is wise, as the user shall understand the task but shall not know too many details that could help him in finding a shot. It was, of course, not allowed to train or pre-configure the own system tuned to the topics. For both interactive and manual search runs, a time limit of 15 minutes was set. It started from the moment the searcher saw the topic until the result set of that topic was returned.

NIST provides suggestions how to conduct interactive experiments [NIST, 2003]. Their design is for measuring the effectiveness of two systems (V_1) and (V_2) using 24 search topics (T_n) and 8, 16 or 24 searchers (S_n). Each user searches 12 different topics. The approach allows the estimation of effectiveness of one system, free and clear of searcher and topic. Statistically, searcher and topic are treated as blocking factors. However, it does *not* solve cross-site comparisons problems.
The final approach is designed of many 2-searcher-by-2-topic latin squares. Table 6.6 shows a 2×2 latin square design.

	T_1	T_2
S_1	V_1	V_2
S_2	V_2	V_1

Table 6.6: 2×2 latin square design

It has to be interpreted in the following way:

- Searcher 1 uses system 1 for search topic 1 and system 2 for search topic 2.
- Searcher 2 uses system 2 for search topic 1 and system 1 for search topic 2.

The performance of searcher S_1 using system V_1 on topic T_1 can be modelled as
$m + s_1 + v_1 + t_1 + e$
(where: m is the grand mean of all performances, s_1 is the effect of searcher 1, v_1 is the effect of system variant 1, t_1 is the effect of topic 1, and e is *error* – the effect of everything else.)
The difference between systems' performance – *treatment effect (x)* – is estimated by the mean of the differences between V_1 and V_2 where the main effects of topic and searcher has fallen out:

$$
\begin{aligned}
x &= \frac{[(m+s_1+t_1+v_1+e)-(m+s_1+t_2+v_2+e)]+[(m+s_2+t_2+v_1+e)-(m+s_2+t_1+v_2+e)]}{2} \\
&= \frac{(t_1-t_2+v_1-v_2)+(t_2-t_1+v_1-v_2)}{2} \\
&= \frac{(2v_1-2v_2)}{2} \\
&= v_1-v_2
\end{aligned}
$$

For covering all 24 topics, the design has to be expanded by replicating the 2×2 square to a 2×24 matrix. The columns are permuted so a searcher completes all tasks on one system. As every search can take up to 15 minutes, every searcher has to do only half the topics. Therefore, the maximum search time for any given user is three hours.
To eliminate the effect of one factor is dependent on the level of another the 2×24 design is replicated in pairs of users to create an 8×24 design.
Table 6.7 shows the design for measuring the effectiveness of two systems (V_1) and (V_2) using 24 search topics (T_n). The table shows the arrangement of the tasks for 8 users (S_n). The design can be repeated with up to two additional sets of 8 searchers. The more searchers are available, the better will be the balance of order related biases. The users are selected randomly as it is the order of the topic presentation [NIST, 2003].

	$T_1 - T_6$	$T_7 - T_{12}$	$T_{13} - T_{18}$	$T_{19} - T_{24}$
S_1	V_1		V_2	
S_2	V_2		V_1	
S_3	V_1			V_2
S_4	V_2			V_1
S_5		V_1	V_2	
S_6		V_2	V_1	
S_7		V_1		V_2
S_8		V_2		V_1

Table 6.7: Measuring the effectiveness of one system

6.4.1 Query Classification

The search query examples are always in a short imperative form like “Find shots of Yasser Arafat” (from the TRECVID 2003 search query collection). They are designed to represent many different sort of queries real users pose: request for video with specific types of people, specific instances of objects, specific activities or locations [Enser and Sandom, 2002]. According to the intent of

the queries, the queries can be classified into four different query classes. As shown in [Yan et al., 2004], these are:

- **Named Person:** Queries for finding a named person. Examples from TRECVID 2003 are: *Find shots of Morgan Freeman* or *Find shots of Pope John Paul II*.
- **Named Object:** Queries for finding a specific object having a unique name. The name distinguishes the object from similar objects. Examples taken from 2003 are: *Find shots of the Tomb of the Unknown Soldier at Arlington National Cemetery* or *Find shots of the Mercedes logo (star)*.
- **General Object:** These queries refer to a general category of objects instead of an specific one. Examples are: *Find shots of one or more tanks* and *Find shots of an airplane taking off.*
- **Scene:** These queries depict a a scene with multiple types of objects which are in a spatial relationship like *Find shots of one or more roads with lots of vehicles* or *Find shots with a locomotive (and attached railroad cars if any) approaching the viewer.*

A classification of topic types from TRECVID 2005 based on [Armitage and Enser, 2005] is provided in appendix A.2.1.
Each query class favours a specific set of features. A useful interrelation is listed in table 6.8.

Query Class	**Useful Feature**
General Object	Shape Colour Motion/Moving Object (Audio Feature)
Named Object	Shape Colour Logo Detection
Named Person	Face Detection Colour Texture (Audio Feature)
Scene	Shape Colour Texture Motion/Moving Object

Table 6.8: Query Classes and useful features

6.4.2 Submissions

Each partner can submit not more than seven runs to NIST. Each run contains one result for every topic, retrieved by a test user. If the user finds more than one result, it is up to the submitting partner to decide which result to submit. The participant partner has to submit a list of at most 1000 shots for each topic in a run. The syntax of the format in which it has to be submitted is defined in a XML schema and can be downloaded from the web[3] [NIST, 2006b].

6.4.3 Test Result Evaluation

The testing and performing assessment is the video shot as defined by the shot boundary reference. The submitted ranked result list of shots is judged manually. All shots from one topic are taken down to some fixed depth in ranked order. This list of unique shots are judged manually based on assessor time and number of correct shots. NIST evaluates each submission to its full depth. Pre-search measures are the average precision and the elapsed time for all runs. Pre-run measures is the mean average precision [NIST, 2006b].
Figure 6.7 illustrates one of their results: A comparison on particular topics of the effectiveness of different systems participating in 2005. The results are presented on the yearly workshop.

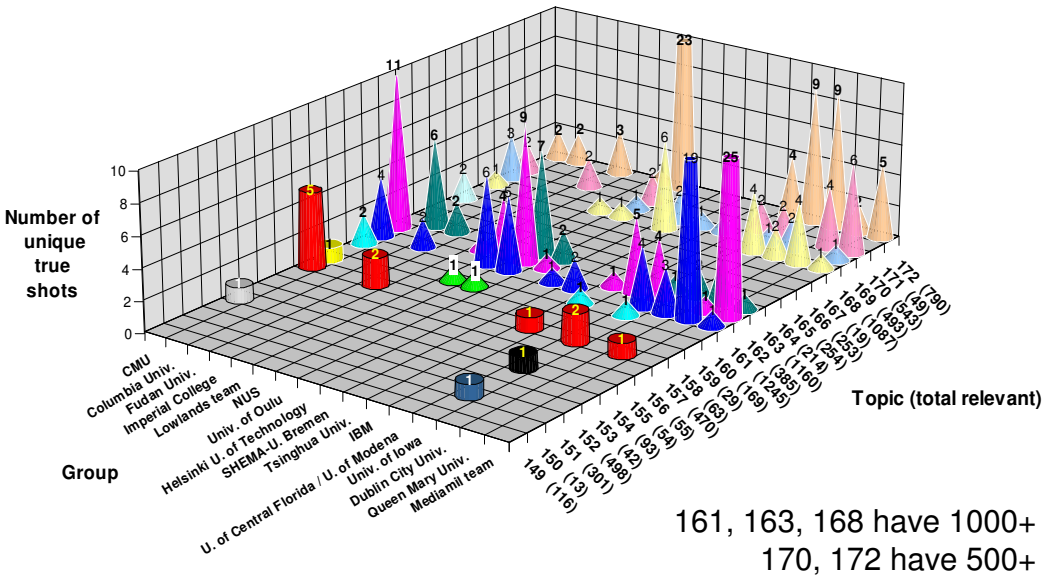

Figure 6.7: 2005: Rel shots contrib. uniquely per topic by team [Over et al., 2005]

[3] http://www-nlpir.nist.gov/projects/tv2005/dtds/videoSearchRunResult.dtd

7 Software Design

> "There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult [Hoare, 1981]."
>
> C. A. R. HOARE
> *British Computer Scientist (born 1934)*

The Software Crisis in the late 1960's [Dijkstra, 1972] leaded to a reflecting how to develop and implement software tools. Computer programs which have been programmed without any documentation became a bigger problem as it was not easy or even impossible to continue or correct them. This was the hour of birth for Software Engineering which utilises the design, use and further development of software systems. Software systems consist of source code and its accompanying documents which are useful and helpful for the usage of the program. Different approaches how to proceed in developing a software system have been introduced. The design of this system is oriented on the Object-Oriented Analysis and Design (OOAD) by Booch [1995]. The process covered with this chapter is divided into a requirements analysis in chapter 7.1 followed by a presentation of use cases and its scenarios in chapter 7.2.

7.1 Requirements Analysis

After making a state-of-the-art analysis (see previous chapters) of existing technologies and detecting basic conditions, a requirements specification can be done. In various discussions between the developer (graduand) and the client (supervisor), both agreed on the following list:

1. The program shall assist in video retrieval.
2. It shall be possible to search, retrieve and playback video shots.
3. The program shall support relevance feedback.
4. The system shall be aligned to the conditions of the TRECVID workshop.
5. The system shall be appropriable in the TRECVID 2006 workshop.
6. The Graphical User Interface shall be easy to understand and simple.
7. The program shall be upgradeable.
8. It shall be programmed in Java.
9. The system shall be documented in an adequate way.
10. Documentation shall be written in English.
11. The system shall be a base system for others built on it.
12. For testing, it will use the TRECVID 2005 development data set.

These requirements have to be attended and fulfilled when realising the program. It will be referred to the listings at adequate position.

7.2 Software Design

After providing the requirements, the next step is to design and specify the software itself. Therefore, it will be traversed step by step from characterisation of some use cases to a scenario description. Beforehand, some interface proposals have been developed. They can be found in Appendix B. Finally, after internal discussion, the interface shall look like the last proposal (figure B.7).

7.2.1 Use Cases

The first step in software design is the alignment of system specific use cases. They describe a functionality or a service of a system, a subsystem or a class. They are useful to give a rough overview to the tasks of the disposed system and the communication with its roles. It is common to visualise these use cases using *Use Case Diagrams* that conform to UML.

Figure 7.1 shows the adapted Use Case Diagram in UML dialect for the planned software and the corresponding annotations of its use cases and actors.

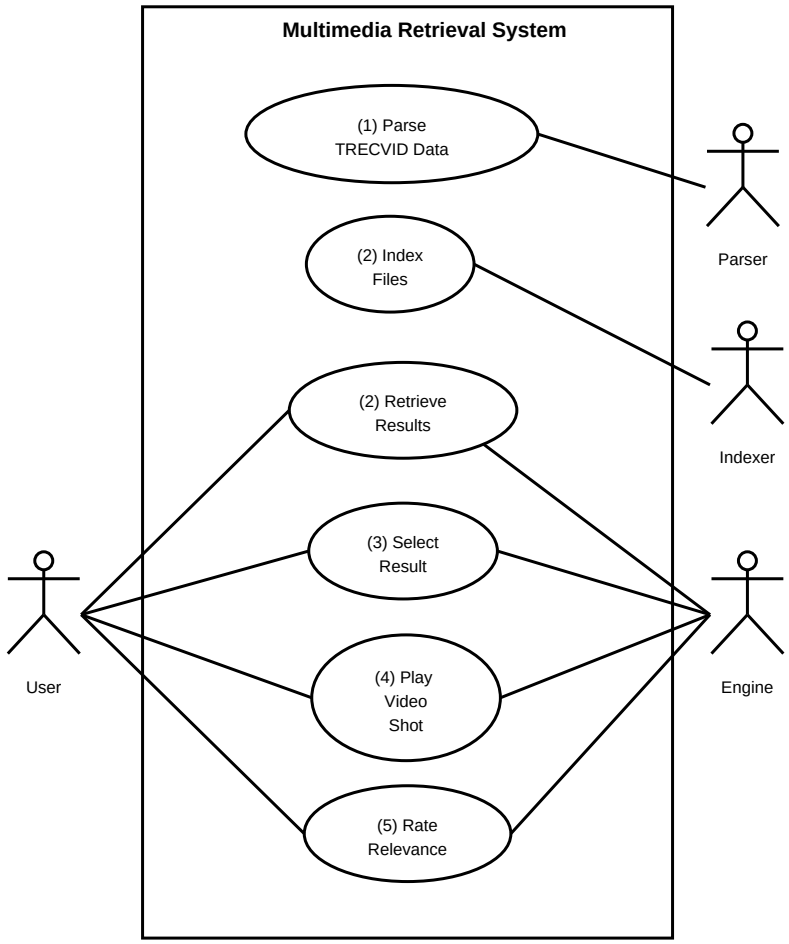

Figure 7.1: Use Case Diagram

- Description of the Actors
  - **User:** This role symbolises the user who works with the software tool
  - **Parser:** The parser parses the TRECVID data set.
  - **Indexer:** The indexer has to create an inverted index file.
  - **Engine:** The engine is the system itself, which will interact with the user.
- Description of the use cases
  - **(1) Parse TRECVID Data:** The parser has to parse the data collection for the indexer and for a later use of the engine and the user respectively. This step has to be done only once. The parsing phase includes both the gaining of the textual video surrogate and the results of the visual feature extraction.

- **(2) Index Files:** The indexer creates the index and the inverted index file. These files have to be created only once.
- **(3) Retrieve Results:** When the user triggers a retrieval, the engine has to detect and list results according to the search query. Therefore, it uses the files created in the use cases (1) and (2). The system includes keyframes to the visual query if they are selected by the user in a previous run (see scenario (6)).
- **(4) Select Result:** When the user selects a keyframe representing a shot from the results, the engine presents detailed information about the shot like date and time of broadcast, title and other available textual information.
- **(5) Play Video Shot:** When the user decides to play the shot that belongs to a keyframe, the engine playbacks the video file. It is also possible for the user to stop and pause the playback.
- **(6) Rate Relevance:** The user can give a relevance feedback which can be used for query expansion and also for a possible visual feature retrieval.

7.2.2 Scenarios

In a next step, the use cases have to be traversed in detail in a scenario description. Here, the first bridges to realisation have to be built by running through different use cases using concrete values. So the later used classes and methods will be indicated. The scenario descriptions are visualised via Sequence Diagrams which are defined in the UML standard [Object Management Group, 2006].

This subsection presents a description using Sequence Diagrams of the in section 7.2.1 introduced use cases.

7.2.2.1 Scenario Description: *(1) Parse TRECVID Data*

Figure 7.2 shows the sequence diagram for the first scenario.
At the very beginning, the parser has to prepare the original data set so that the engine can retrieve all necessary data as fast as possible. This step has to be done only once. Therefore, two parsing sequences are necessary: One to parse the data into a handy format for the engine and a second sequence to parse the data into a format that is suitable for the indexer. Both parsing sequences work the same way.
This scenario deals only with the parsing into the handy format for the engine. The relevant

methods for the second parser are written in brackets.

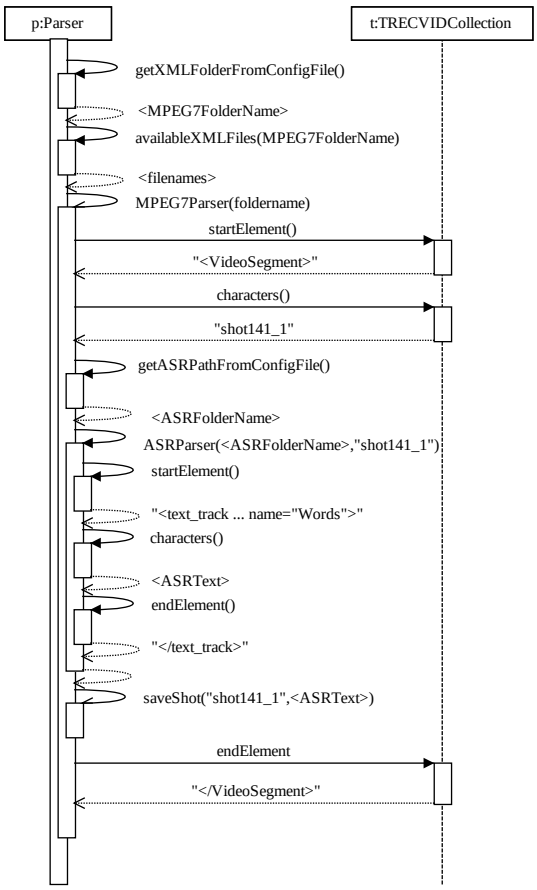

Figure 7.2: Sequence Diagram for *(1) Parse TRECVID Data*

Before the parser can start, he has to find out where the TRECVID data set can be found. This information is stored in a config file called `parser.cfg` which can be found in the root directory. To detect it, the parser calls the method `getXMLFolderNameFromConfigFile()` and stores the `foldername`. Therefore, it detects all available xml files in that folder using the method `availableXMLFiles(foldername)`.
After this set up, the system can start the parsing activity.
At first, it calls `MPEG7Parser(foldername)` (`MPEG7IndexParser()`). This method goes through the XML files and checks for the XML tag `"<VideoSegment>"`. When it appears, the method `startElement()` is triggered. After that, the method `characters()` is triggered that returns `"shot141_1"`. This is the id of the video segment.
(Dependent on the XML tag, other elements like the *MediaTimePoint* or the *MediaDuration* are

detected on the same way. These tags are ignored in this scenario description to keep it simple and understandable.)
Afterwards, the parser has to detect the `<ASRFolderName>` out of the config file using `getASRPathFromConfigFile()`. Then `ASRParser(<ASRFolderName>,"shot141_1")` (`ASRIndexParser()`) is triggered which parses the ASR XML file in the same way. It searches the XML tag `text_track` and returns the `<ASRText>` which belongs to this specific shot in dependency of the calculated *MediaStartPoint* and *MediaEndPoint*. Finally, the parser stores the parsed information using the method `saveShot("shot141_1", <ASRText>)` and ends when the XML tag `"</VideoSegment>"` appears.

7.2.2.2 Scenario Description: *(2) Index Files*

Indexing the parsed files is a procedure that also has to be done only once. For this step, the Terrier system [1] which is provided by the Glasgow IR Group is used as indexer. Terrier is a Java based framework for the rapid development of large-scale information retrieval applications and provides indexing and retrieval functionalities. It includes the ability to index the standard TREC collections [Ounis et al., 2005, Information Retrieval Goup, 2005].

Indexing is a process in which keywords are assigned to available documents. Different steps are involved in this:

- *Lexical Analysis* (Intra document parsing and Tokenising): In this process, a stream of characters of a document is converted into a stream of words. These words are the candidate word which might be adopted as index terms.
- *Stop-word removal*: Words that appear too frequently in the documents are bad discriminators. Therefore, they have to be eliminated as later index terms. This step reduces the size of the indexing structure considerably.
- *Stemming* (removal of affixes): In this process, all words with the same roots are minimised into the same root. A stem is the part of a word which is left after the removal of all affixes. An example: *connect* is the stem for the variants *connecting*, *connections*, and *connected*.

A more detailed description of these basics can be found in [van Rijsbergen, 1979, Belew, 2000]. The result of this indexing is an index matrix: $Index\colon\ doc_i \xrightarrow{about} \{kw_j\}$. To speed up retrieval in the index, Terrier inverts all documents into a big index. This inverted index file is a document-term matrix representation. Rows become columns and columns become rows:
$Index^{-1}\colon\ \{kw_j\} \xrightarrow{describes} doc_i$

[1] http://ir.dcs.gla.ac.uk/terrier/

For the visual retrieval, it is necessary to process a visual feature extraction. The extraction was done by Dr. Xavier Hilaire, a precise description of his proceeding does not exist yet. He provides:

- colour histograms
- textures
- dominant colours
- edge histograms
- contour shapes

7.2.2.3 Scenario Description: *(3) Retrieve Results*

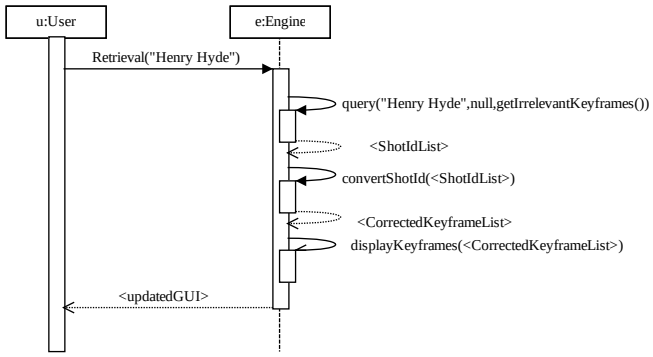

Figure 7.3: Sequence Diagram for *(3) Retrieve Results*

The user wants to find shots of U.S. Congressman Henry Hyde's face, whole or part, from any angel. (This example is taken from TRECVID 2004 search task.) Therefore, he enters the search query "Henry Hyde" and pushes the search button. So, he triggers the method `Retrieval("Henry_Hyde")`. The search query is a String parameter. First, the system checks if there are any keyframes rated relevant by the user it has to add to the search query. Here, it is not the case. So, the engine calls the method `query("HenryHyde",null, getIrrelevantKeyframes())`. This method goes through the data set and returns a list of all keyframes that are associated to the string "Henry Hyde" and returns a list containing the *shot id*. As the returning keyframe list contains only a relative path to the retrieved keyframes, the method `convertShotId(<ShotIdList>)`. completes the full path to each keyframe. Afterwards, the engine calls `displayKeyframes(<CorrectedKeyframeList>)` which

displays all retrieved keyframes in the destined panel of the GUI. Thus, the user maintains an updated GUI showing the keyframes.

7.2.2.4 Scenario Description: *(4) Select Result*

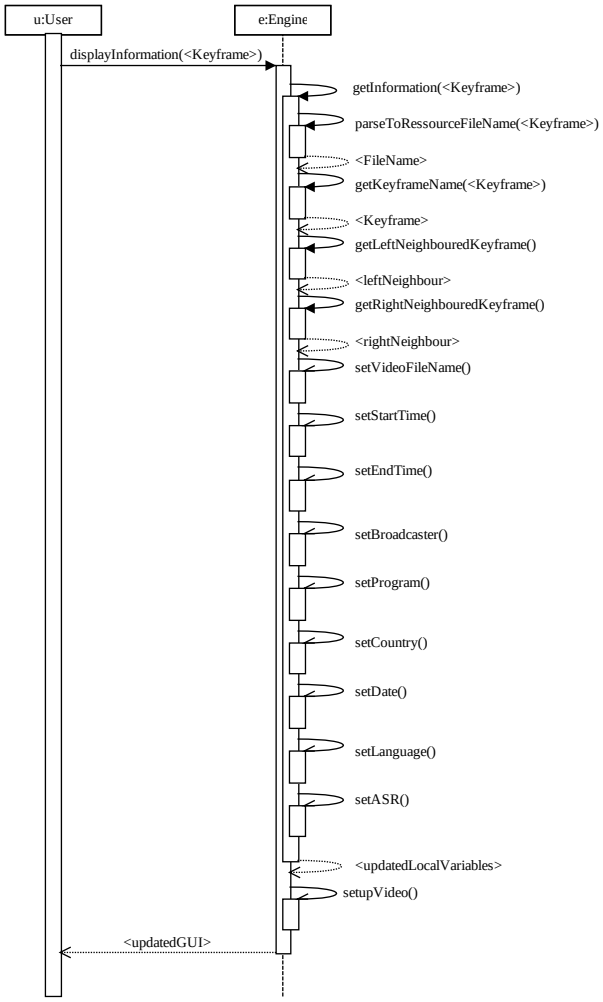

Figure 7.4: Sequence Diagram for *(4) Select Result*

The user calls the method `displayInformation(<Keyframe>)` after clicking on a keyframe. Then, the engine triggers the method `getInformation(<Keyframe>)`. This method first calls the method `parseToResourceFileName(<Keyframe>)` to achieve the name of the parsed text file (see scenario one) that belongs to this keyframe. After that, it collects all available information about the video shot which is symbolised by the keyframe:
`getKeyframeName(<Keyframe>)` returns the path to the selected keyframe. Both

`getLeftNeighbouredKeyframe()` and `getRightNeighbouredKeyframe()` calculate the names of the neighboured keyframes. Metadata information is read from the parsed file and stored in local variables using the methods `setVideoFileName()`, `setStartTime()`, `setEndTime()`, `setBroadcaster()`, `setProgram()`, `setCountry()`, `setDate()`, `setLanguage()` and `setASR()`.
Finally, the engine displays all attained information, starts the shot in the video file using `setupVideo()` and the user maintains the updated GUI.

7.2.2.5 Scenario Description: *(5) Play Video Shot*

The user has different chances to access video files: He can start a video, pause a video and jump to every time point of the video file using a slider. Furthermore, he has the possibility to gain more information about the video that is currently played like Media Location, Content Type, duration and current position. Additionally, he can change the volume of the audio output.
These features are automatically supported by the Java Media Player that is part of the Java Media Framework. [2] A detailed description of its internal methods can be found in the API. [3]

7.2.2.6 Scenario Description: *(6) Rate Relevance*

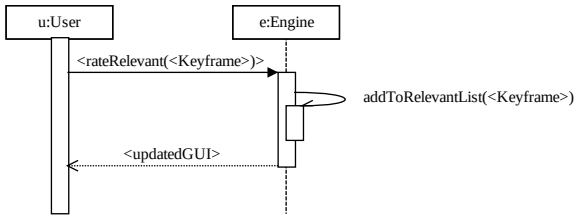

Figure 7.5: Sequence Diagram for *(6) Rate Relevance*

The user can rate a video shot/keyframe in clicking on the radio button under the keyframe. In the example, the user decides to rate a result as “relevant”. There upon, the engine adds the keyframe to the list that contains all relevant keyframes using the method `addToRelevantList(<Keyframe>)`. Finally, the user gets an updated GUI.

[2] http://java.sun.com/products/java-media/jmf/
[3] http://java.sun.com/products/java-media/jmf/reference/api/index.html

8 Implementation and Documentation

> "We view the documentation as being (at least) as important as the product itself: if there is good documentation, a software product can be revised or replaced relatively quickly; without good documentation, software products are of questionable long-term value [Parnas and Madey, 1995]."
>
> DAVID L. PARNAS
> *American pioneer of software engineering (born 1941)*

A good documentation of a developed software system is mandatory, as it helps others in understanding the structure and the source code of the system. This chapter offers a closer look at the structure of the software which fulfils the requirements number (9) and (10). After a short overview in chapter 8.1, chapter 8.2 explains the requirements and infrastructure for the system here at Glasgow University. Chapter 8.3 presents more technical details about the developed parser while chapter 8.4 presents details about the multimedia retrieval tool.

8.1 Overview

As shown in the use cases before, the challenge to the new retrieval system is divided into three pieces. At first, there is the necessity to create a system that parses the available TRECVID data set and saves the result in a form that is acceptable for fast retrieval. The second task is to index the data set. Finally, the user shall have a graphical user interface which he can use.

As IR Group already provides the Terrier system, it was suitable for the indexing part of it. Both

parser and the actual retrieval system had to be designed and implemented. The implementation details will be raised in the following section.

8.2 System Environment

Both systems are trained on the TRECVID 2005 video data set. (see requirement number (12).) A brief overview of the collection has been given in chapter 6.2. The 2005 data collection is stored on `\\Mota\kspace\trecvid-2005` and can be accessed in the university network. The tools are suitable for collections to come without the necessity to change source code (see requirement number (7)).
Due to requirement number (8), the programs are implemented in Sun Java 5.0. They are stored in a subversion repository on `https://ouen.dcs.gla.ac.uk/repos/MIR/kspace` and can be accessed by the group members.

8.3 The Parser

To understand a software system, it helps to describe the structure of it. A textual specification of the classes and its methods is appended to the source code in HTML format (created with Javadoc utility). The structure of the software system is oriented on the Filesystem Hierarchy Standard [Russell et al., 2004]:

- **/bin:** Holds all compiled binary files.
- **/doc:** Contains the API documentation in HTML format which is extracted from the Java Source Code. The documentation specifies the classes.
- **/src:** Contains the implemented source code.

The root folder contains a config file (`parser.cfg`) where the user can set the path to the MPEG-7 and ASR files of the TRECVID collection without changing the source code. Besides, he can set the output folder where the system will store its parsing results.

In Java, there are two standard approaches to parse XML files.

1. *DOM* – creates a tree in which you can navigate with various methods
2. *SAX* – creates events for the start and the end of an element. It triggers callback methods to handle them.

According to CollabNet [2005], the different existing open-source and commercial libraries for binding XML data to Java classes have a deviant performance. As the data collection is a large set, it is even more important to find the best performing way to access the XML files. Therefor, the *Simple API for XML* (SAX) appears to be most promising. It is the *"fastest and least memory-intensive mechanism that is currently available for dealing with XML documents"* [Sun Microsystems, 2006].
As the parser implements SAX, it handles the XML information as a single stream of data. The data stream is unidirectional, so that previously accessed data cannot be read again without restarting the parsing. No external libraries are necessary as SAX is part of the used Java Developing Kit. More information about SAX can be found in [Means and Bodie, 2002].

8.3.1 Parsing Output

The parser produces two different kind of output files:

1. The parse results for the Terrier System.
2. The results for the retrieval system.

As Terrier needs input in official TREC format for indexing, the documents of parsing cycle (1) are in the SGML style markup. The format is

<DOC>
<DOCNO> document number </DOCNO>
<DATE> date </DATE>
<DESC>document text</DESC>
</DOC>

Having a closer look at this format, each shot has a <DOCNO> tag including the video identifier string and the shot identifier. The document number is the assembled *video id* and the *video segment id*, separated by a "/". So the document number is always in the format *TRECVID*<*year*>_<*number*>/*shot*<*number*>_<*number*>. This information is taken from the MPEG-7 files of the official collection. Then, it contains the <DATE> tag which includes the broadcasting date of the shot. The document text is surrounded by the <DESC> tag and is taken from the ASR file after calculating the duration of each shot using the MPEG-7 file.

Table 8.1 shows an extract of parsing cycle (1). Each video file has an associated text file in ASCII format.

```
<DOC>
<DOCNO> TRECVID2005_141/shot141_1 </DOCNO>
<DATE> 01_03_2005 </DATE>
<DESC>
New
world
of
</DESC>
</DOC>
```

Table 8.1: Parsing result (1) in SGML format

The resulting documents of parsing cycle (2) contain more information. It produces files named after the *video segment id* for each shot in ASCII format. The files are stored in folders named after the *video id*.
Figure 8.2 lists an example file.

```
VideoID: TRECVID2005_141
MediaUri: 20041030_133100_MSNBC_MSNBCNEWS13_ENG.mpg
VideoSegmentId: shot141_1
MediaTimePoint: T00:00:00:0F30000
Starting FrameNumber:0
Startsecond: 0
Media Duration: PT00H00M02S15075N30000F
Media Duration in seconds: 2
Media Duration FrameNumber: 75
Keyframe Name: shot1_1_RKF
Broadcaster: MSNBC
StartTime of Broadcast: 1114718160
Program: MSNBC News
Country: United States
Date: 2005-03-01
Language: US English
ASR Text:
New
world
of
```

Table 8.2: Parsing result (2)

8.4 The multimedia retrieval tool

The graphical user interface is realised using the GUI toolkit Swing. To set up the Look and Feel, it uses the libraries JGoodies Looks Version 2.0.1[1] which are provided under the terms of the BSD open source license. It is designed to fulfil requirement number (6).
The structure of the software system also is oriented on the Filesystem Hierarchy Standard:

- **/bin:** Holds all compiled binary files.
- **/doc:** Contains the API documentation in HTML format which is extracted from the Java Source Code. The documentation specifies the classes.
- **/etc:** Contains the configuration files for the Terrier System.
- **/lib:** Contains compiled Java classes that are necessary to run the program.
- **/log:** Contains the logfiles used for evaluation.
- **/share:** Contains the stop word list for the Terrier system.
- **/src:** Contains the implemented source code.
- **/var:** Contains the inverted index file.

The root folder contains the configuration file `mir.cfg` where the user can change the path to the data set. The video surrogates are the output files of the parser. The visual features for each keyframe are stored in ASCII format. The whole system is aligned to the conditions of the TRECVID workshop (see requirement number (4)). Hence, it can be used for the TRECVID 2006 workshop (see requirement number (5)).
The actual retrieval system is divided into two pieces. There is the graphical user interface as front end and the retrieval engine working in the back office. The code is seperated into several Java packages: The *terrier*, *uk.ac.gla.terrier.querying* and *uk.ac.gla.terrier.structures* packages contain classes for the Terrier system. The *trecvid* package contains classes and config files for the visual indexing part. *trecvid.clustering* is for clustering results. The package *trecvid.data* contains classes for handling the data set. The *trecvid.engine* package contains all classes that are used for the retrieval. The *trecvid.evaluation* packages provides classes for evaluation. The *trecvid.gui* package contains all layout information and elements to display the Graphical User Interface (see requirements numbers (1) and (2)) and to support relevance feedback (see requirement number (3)). There is a constant data flow between the packages.

[1] `http://www.jgoodies.com/downloads/libraries.html`

8.4.1 Graphical User Interface

After making a state-of-the-art analysis (see section 6.3) as basis for discussion, different graphical user interfaces were proposed. The suggestions can be found in Appendix B. Important condition was of course to fulfil requirement (6). Finally, the last proposal (figure B.7) was accepted as the most useful solution.

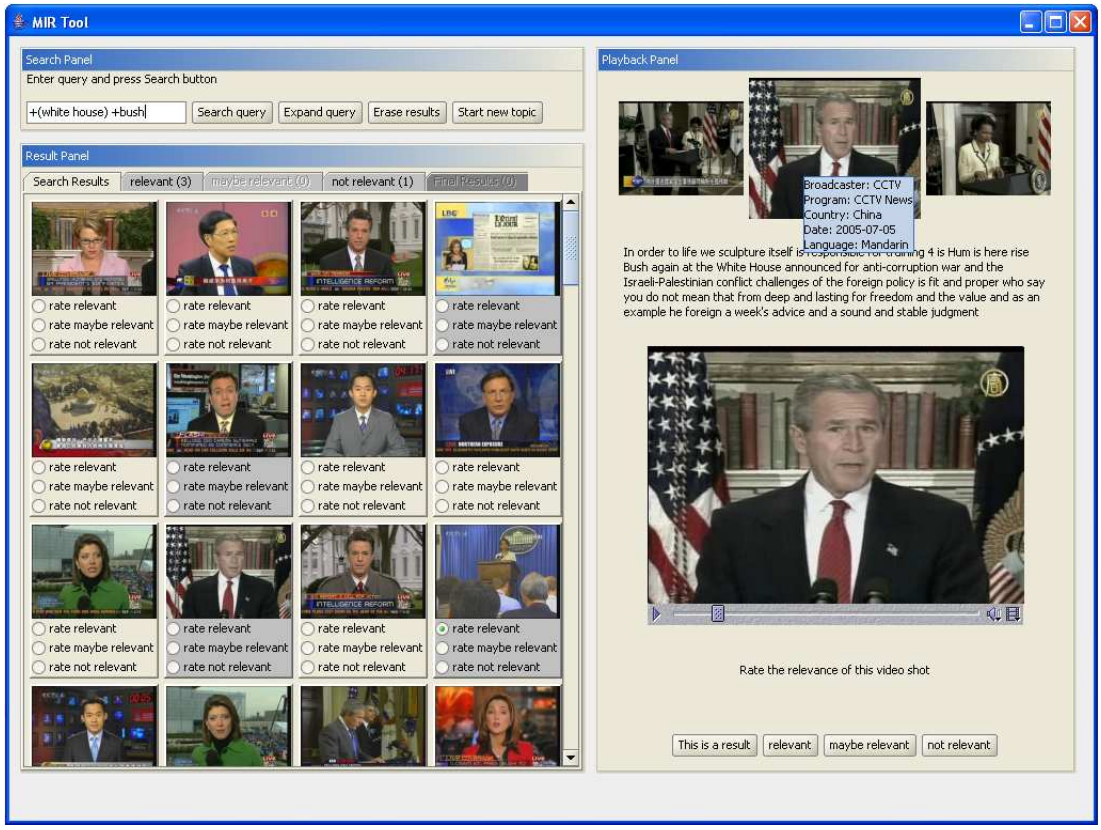

Figure 8.1: Graphical User Interface

Figure 8.1 shows a screenshot of the developed system. It can be divided into three parts: The *Search Panel*, *Result Panel* and the *Playback Panel*.
They will be introduced in the following subsections.
Before the user can use the system, he has to log on to it and enter his *user id*, the *run id* and select the *search topic* he wants to perform. This is necessary for the right assignment of the logfiles that are used for the later evaluation. For this procedure, a small window pops up on startup.

8.4.1.1 Search Panel

The *Search Panel* contains a text field for entering the query. The textual retrieval engine Terrier has an advanced query language [Ounis, 2005]. The query syntax considers the boolean algebra. For combining query words with the boolean OR, the words have to be written with space between each word. The boolean AND can be entered in adding the + symbol at the beginning of each query word. Words beginning with the – symbol will be ignored (boolean NOT). Proper search queries would be e.g.:

- t_1 t_2: retrieves entries with either t_1 or t_2
- t_1^2.3: the weight of t_1 is boosted to 2.3
- +t_1 -t_2: retrieve entries with t_1 but not t_2
- +(t_1 t_2): both terms t_1 and t_2 are required
- *field:*t_1: retrieves entries where t_1 appears in the specified field (date or desc)

Combinations like "+t_1 +t_2 -t_3" are possible.

The retrieval can be triggered by pressing *Enter* or by clicking the "search" button. Clicking the button "Expand query" will open a new window for relevance feedback. Read more about it in chapter 8.4.1.4. The button "Erase everything" will remove former results as relevance feedback. Before removing them, the system will ask the user to confirm this decision. For starting a new TRECVID search topic, the user can press the "Start new topic" button. All former results will be erased and a new window will pop up where the user can enter a new *user id*, *run id* and select the next *search topic*.

8.4.1.2 Result Panel

The *Result Panel* is divided into five tabs. The *Search Results* tab lists all retrieved video shots ranked using the PL2 model [Amati and van Rijsbergen, 2002]. Each retrieved shot is represented via the extracted keyframe. When clicking on one keyframe, the video shot and more information will be displayed in the Playback Panel (see following subsection). Under each keyframe, the user can click on radio buttons to rate the relevance of that particular result. According to their rating (relevant, maybe relevant and not relevant), the keyframes will be displayed in one of the other three tabs (*relevance tabs*). Keyframes which have been retrieved in a prior retrieval are displayed in another colour for a better identification. Empty relevance tabs are disabled by default. The number of rated entries is displayed in each title of the tabs. Results can be moved to other tabs by

rating them again. The features of the keyframes that are rated *relevant* will be proposed as visual query for the next search in the query expansion window (*explicit relevance feedback*). The fifth tab (*Final Result* tab) contains the keyframes that the user considered to be a result for the current search topic.

8.4.1.3 Playback Panel

When the user decides to play one video shot, he gets everything displayed in the *Playback Panel* which is placed on the right-hand side of the graphical user interface. On the top, he sees the selected keyframes in its context – with its neighboured keyframes to the left-hand and the right-hand side. He can obtain additional information about the video (Broadcaster, Program, Country, Date and Language) in moving the mouse over the keyframe (see figure 8.1). When clicking on the neighboured keyframes, the Playback Panel will be updated displaying the video shot and the additional information.

Figure 8.2: Graphical User Interface: Add text to query

Underneath these keyframes, the interface displays the automatic speech recognition text of the selected video shot. Here, the user can mark text and add it to the original search query in pressing the right mouse button (on Apple Macintosh machines: Ctrl. and mouse click) (see figure 8.2).

In the middle of the Panel, the video shot is played. When the shot ends, the video pauses. The user can start and pause the video anytime on clicking on the typical icon under the video. The current playing position is presented with a slider bar. The user can use this bar to navigate in the video file. Furthermore, the user can change the volume and read the Media Properties on clicking on the representative icons. Then, a new window pops up which shows additional information like the name of the video file, the duration and the current position (see figure 8.3).

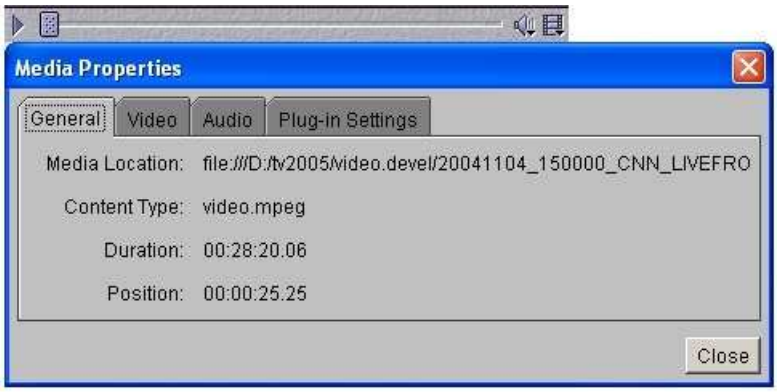

Figure 8.3: Graphical User Interface: Player Interface and Media Properties

On the bottom, the user can either mark a shot as a result or rate the relevance of the shot via buttons. Clicking on one of the four buttons will determine the time stamp of the shot that is currently played, detect the name of the shot in the MPEG-7 file and update the Result Panel. Every played shot is automatically added to the candidate list for the visual query visualised in the query expansion window (*implicit relevance feedback*).

8.4.1.4 Query Expansion Window

The query expansion window assists the user to refine his query. Figure 8.4 shows a screenshot of it. On the top, the panel displays all keyframes the user marked as relevant or he played in this run. He can select or unselect each keyframe and indicate by this means whether he wants to add it as visual query or not. The rated keyframes are selected by default.

In the middle of the panel, he can set a time span if he wants to confine the search according to a date. The system also proposes exact dates, implicitly ascertained from the videos played before. Selecting this option will update the textual query. Clicking on a specific date will implement the field option of the Terrier system, e.g. DATE:21_12_2005. The date option is disabled by default. On the bottom, the system suggests query terms that can be added to the query. The terms are taken from the video surrogate of the relevant rated or clicked keyframes or – if no keyframes have been rated or clicked before – from the Top 100 results of the initial query (*pseudo relevance feedback*). The user can change or add new terms and specify for each term if it *has* to appear (AND), if it *may* appear (OR), or if it *may not* (NOT) be in the video surrogate. Besides, the user can change the weight for each term.

Figure 8.4: Query Expansion Window

9 Evaluation

> "To do successful research, you don't need to know everything, you just need to know of one thing that isn't known [Salisbury, 1999]."
>
> ARTHUR SCHAWLOW
> *American physicist (1921–1999)*

The developed system can be the base for various research in the field of video retrieval. One research question is presented in chapter 9.1. Chapter 9.2 presents the result of a simulated user study. Chapter 9.3 explains the setting for a TRECVID user study. Questionnaires for evaluation are introduced in chapter 9.4. Chapter 9.5 explains the common experimental procedure.

9.1 Experimental Hypotheses

The developed system supports explicit relevance feedback and rudimentary implicit relevance feedback: The user has to rate the relevance of a shot, can select between proposed query terms and can mark a shot as a final result. Interesting would be to find out, how much influence implicit relevance feedback can have on video retrieval. An adequate hypothesis is: *"A combination system of implicit and explicit features is better than the system based on explicit feature only for video retrieval"*. For evaluating this, two systems S_1 and S_2 with different forms of interface support for facilitating the use of relevance feedback have to be compared in a user study. The so far presented system can be used for as S_1, a second system supporting more implicit relevance feedback has been developed based on it. It includes some ideas for implicit relevance feedback as listed in the following:

- A click on keyframe indicates interest in it.
- The duration of video playing time indicates maybe relevant content. The longer a video is played, the higher the likelihood that it is relevant.

- (Almost) neighbouring shots rated relevant indicate importance of shots between them. The enclosed keyframes might be part of the same story.
- The multiple appearance of the same date for different shots indicates the importance of the date because of a time limited event. Searching e.g. for “Germany football world cup”, most results might be found in videos broadcast in June 2006.
- Play/pause video and usage of slider indicates interaction with the video. The more interaction, the more attention a user spends on a video.
- Copying terms from ASR indicates relevance of the text.
- Looking at the video metadata (Java tooltip on the main keyframe) signifies interest in its content.

As the user might give the implicit relevant feedback unconsciously, it has to be considered very well how to judge this feedback. Kelly and Belkin [2004] e.g. performed a user study on the relevance of display time as implicit feedback (in a textual retrieval system). They concluded that there is no general relationship between display time and usefulness. Other studies [Claypool et al., 2001] found out that users display documents that they find useful longer than those they do not. So, experimental results deviate.
The different relevance features must be weighted for classifying the importance of a result. If more actions appear on the same result, the weighting must grow, as the implicit factor grows as well. Low-level feedback information such as clicking on a keyframe or looking at the metadata cover a low weighting span.

Feature	**Weighting**
Click on keyframe	10
Playing duration $> 1sec$	10
Playing duration $> 2sec$	20
Playing duration $> 3sec$	30
$>$ 2 interactions	10
$>$ 3 interactions	15
$>$ 4 interactions	20
looking at metadata	5
copying terms	5
neighboured shots rated relevant	20
date appeared before	20

Table 9.1: Possible weighting of implicit features

The weighting can increase, e.g. dependent on the time, a video is played. Explicit feedback has the maximum weighting of 1.0. Table 9.1 list a proposal on how to weight the different features. The features can be arranged into three categories:

- C_1: Click on keyframe
- C_2: View of keyframe (“Playing duration”)
- C_3: Interaction with keyframe

As implicitly detected results may not receive a higher weighting than explicitly selected, the significance of implicit feedback can be combined to a value between 0.0 and 1.0. So, the implicit feedback must be aggregated in a strictly monotonic increasing function with values between 0.0 and 1.0 . A possible function to achieve this aim is $f(x) = 1 - \frac{1}{x}$, where $\{x \in \mathbb{R} | x \geq 1\}$. Figure 9.1 shows a plot of the function.

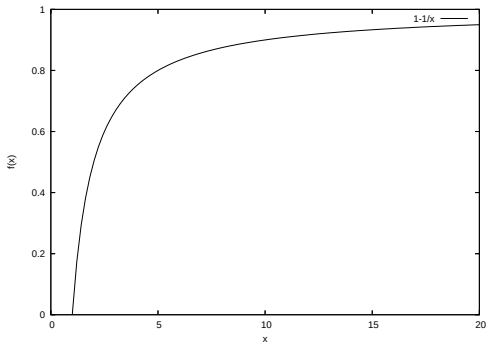

Figure 9.1: Plot of the proposed function

Using more implicit relevance feedback, the formulation and execution of new search queries has to be considered again. In the developed system, the user decides when he wants to start a new retrieval and which details he wants to add to the search query. This concept should be changed as the user is not always aware of the feedback he gives. Different scenarios are imaginable:

- The system automatically formulates and executes a new query after *X* interactions.
- The user is asked explicitly for starting a new query.
- The user can start a new retrieval (e.g. in using a button *Check again*).
- The system can automatically update a *related video shots* window.

9.2 Simulated Experiment

For testing this hypothesis on the developed systems, two different test runs have been carried out. Before running real user tests, the user behaviour was simulated. The first test simulated a searcher using the system S_1 to perform the 24 TRECVID topics from 2005. An initial query was given to the retrieval engine and after a first retrieval, the first five relevant results were taken for use in automatic query expansion. (The relevant shots were detected by comparing the retrieval results with the content of the file `search.qrels.tv05` which was provided by NIST for evaluation purposes. It contains a list of all relevant shots for each search topic.) The idea behind this is that a user would click only on those results which appear to be relevant. The retrieval is then started again with an updated query (with a maximum of six terms – the top six terms that were detected so far) and again, the top five new results which have not been considered before are used as source for a query expansion. These steps were repeated up to 13 times.

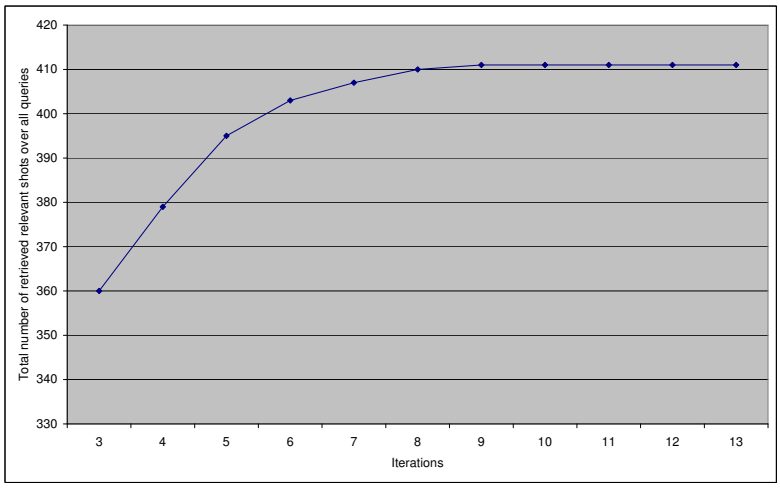

Figure 9.2: Total number of retrieved relevant shots over all queries (S_1)

As illustrated in figure 9.2, the total number of retrieved relevant shots over all queries increases in the first iterations. It also shows that after nine iterations, the maximum number of results are retrieved with no new results retrieved in subsequent iterations.

This result is also sustained by the mean average precision of the simulated test runs. As illustrated in figure 9.3, the precision increases at the beginning, but the more iterations, the worse it gets. This can be explained by the increasing number of irrelevant results that are added in later iterations.

For evaluating the system S_2, a user giving random implicit feedback was simulated. In each iteration, the detected terms from query expansion belonging to the first five retrieved documents were weighted randomly based on the proposal of table 9.1. A user behaviour was modelled using

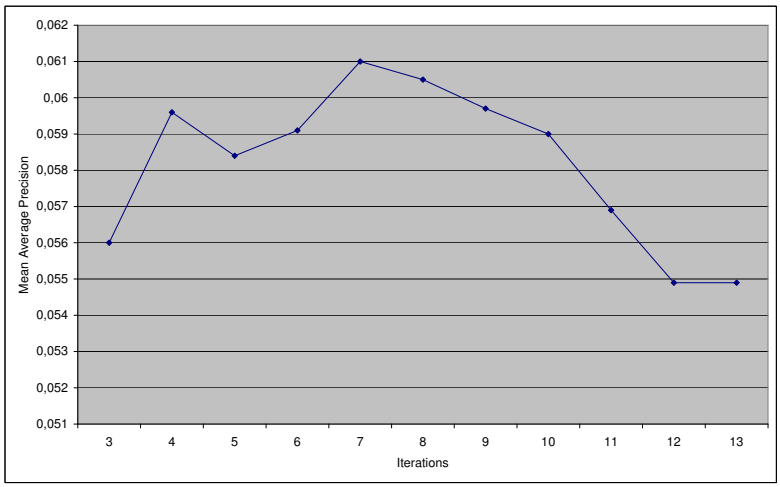

Figure 9.3: Mean Average Precision (S_1)

$C_1 + C_2 + C_3$, where $C_1 < C_2 < C_3$. Possible simulated behaviours are e.g. “Click on keyframe” (which adds the weighting of 10 to the retrieved terms) or “Click on keyframe” and “View of keyframe” (which adds the weighting of 30 (=10+20) to the retrieved terms). As a refined query consists of the top six weighted terms, the simulated user behaviour influences the new query implicitly.

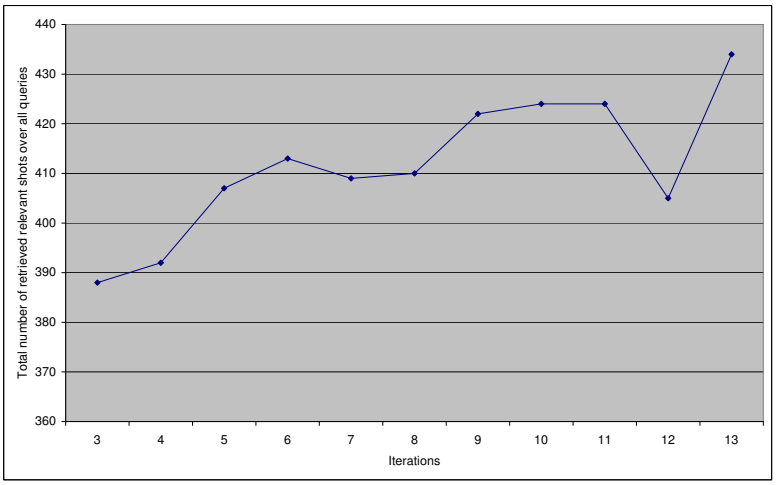

Figure 9.4: Total number of retrieved relevant shots over all queries (S_2)

Deviating from the results of S_1, the total number of retrieved relevant shots over all queries of S_2 tends to result in – apart from few deviations – more results (see figure 9.4). The hypothesis that “a combination system of implicit and explicit features is better than a system based on explicit feature only for video retrieval” is supported by the mean average precision, which stays on the

same level during all iterations (see figure 9.5).

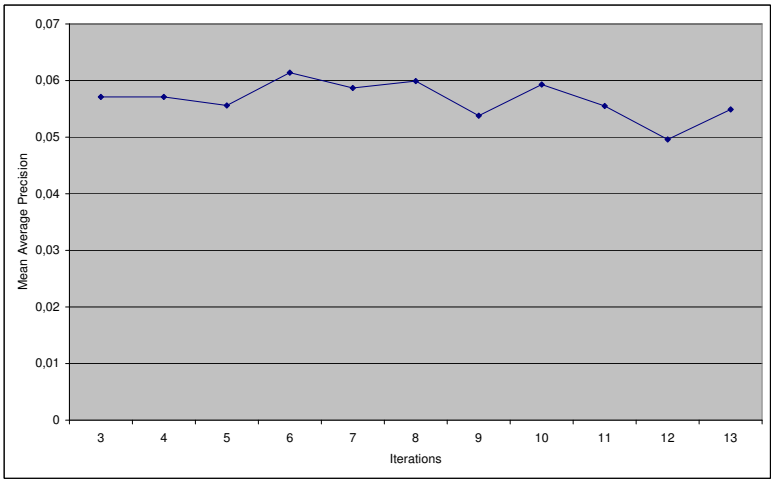

Figure 9.5: Mean Average Precision (S_2)

Summarising these simulated test results, the claimed hypothesis can be validated: The simulated combined system S_2 returned better results than the system S_1 based on explicit features only.

An interesting research question is: *"Which weighting factors for the different feature categories are appropriate for feedback?"* Therefore, another two runs using a system S_3 and S_4 were conducted which simulate random user behaviour under consideration of $C_1 = C_2 = C_3$ (S_3) and $C_1 > C_2 > C_3$ (S_4) respectively. S_3 gives an equal weighting factor of 10 for each feature while S_4's weighting is based on table 9.2.

Feature	**Weighting**
Click on keyframe (C_1)	30
View of keyframe (C_2)	20
Interaction with keyframe (C_3)	10
looking at metadata	5
copying terms	5
neighboured shots rated relevant	5
date appeared before	5

Table 9.2: Weighting of implicit features (for S_4)

As figure 9.6 illustrates, S_4 retrieved a higher number of results than both S_2 and S_3, so a model should weight $C_1 > C_2 > C_3$.

The result is also supported by figure 9.7 which illustrates the precision after 10 shots retrieved.

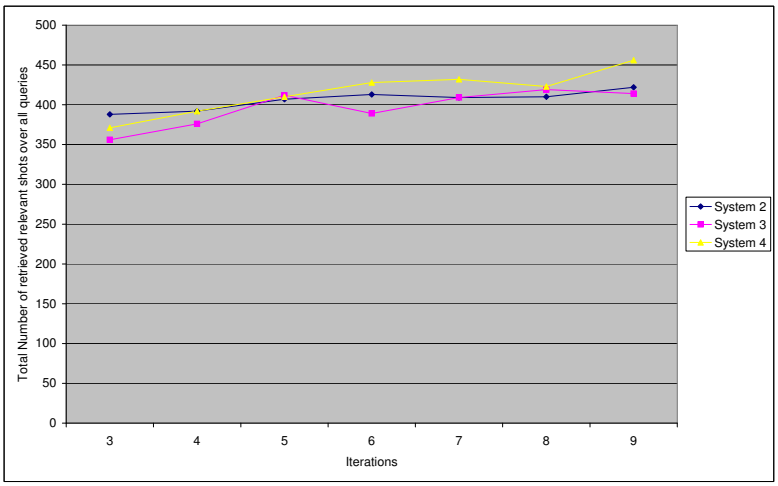

Figure 9.6: Total number of retrieved relevant shots over all queries (S_2, S_3 and S_4)

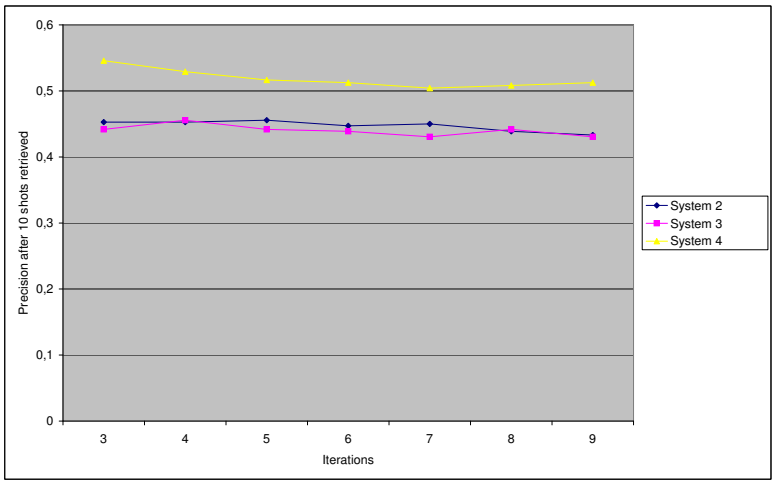

Figure 9.7: Precision after 10 shots retrieved (S_2, S_3 and S_4)

9.3 Experimental Setup

For testing the two systems under the conditions of the 2005 TRECVID workshop [NIST, 2006b], at least eight interactive runs have to be conducted. Experimental design principles such as randomising the order in which topics are selected for each run to balance learning effects must be considered.

Before starting a user experiment, each user receives at least a 10-minute tutorial on how to use the system. Conforming with the 2005 guidelines [NIST, 2006b], 15 minutes are allocated for every user per search task. This time includes the time needed for reading the TRECVID topic. The

guideline outlines the experimental process to be followed. Both users and search topics should be arranged using the 8-seacher-by-24-topics latin square design (see table 6.7) as explained in chapter 6.4.

9.4 Questionnaires

For collecting data about user satisfaction and characteristics, Smeaton and Wilkins [2004] created a set of questionnaires that can be used for doing interactive search experiments. They suggest three separate questionnaires:

- A *pre-experiment user questionnaire* should be completed by the user before starting the training of the system.
- A *post-topic questionnaire* should be completed by each user after completing a topic.
- A *post-experiment questionnaire* should be completed by each user after finishing the whole experiment.

Their questionnaires contain three kind of question:

1. Likert scales
2. semantic differentials
3. open-ended questions

The five point *Likert scale* technique is taken for quantifying the expression of agreement or disagreement of a user. It presents a set of attitudes. For measuring the level of agreement, a numerical value from one to five is used. The value can be measured in calculating the average of all received responses.
The other type of structured question, the *semantic differentials* provide a set of bipolar adjectives with a five-step rating scale between them. The adjectives can express one's attitudes.
Open-ended questions are useful to find out more about the reasons, why a subject behaves the way he/she does and provides the chance to give free comments on aspects of the system.
These and other survey techniques are presented summarised by the Human/Computer Interaction Laboratory [2006] of the University of Maryland/USA.

9.5 Experimental Procedure

Before beginning the actual experiment, each user should complete the *pre-experiment user questionnaire*. After logging on and accomplishing a test search, the user starts the topic search. All his actions will be logged automatically for evaluating (see chapter 9.5.1). The user has to fill in the *post-topic questionnaire*. After finishing the search topics, he/she has to fill in the *post-experiment questionnaire*. The participants should also have the chance to provide comments about the system at the end of their run.

9.5.1 System Logging

While running a search topic, the system automatically logs the user's results, his actions and related information generated by the system.
After each topic, the user results are stored in two different files: One log file in XML format is the file that has to be submitted to NIST for evaluation. The second one can be used for evaluation using a tool provided by NIST. [1] According to [NIST, 2006b], they shall contain at most 1000 shots. Examples listing can be found in appendix C.1.1 and C.1.2.
During the interaction, the user behaviour is logged. These log files are named based on the subjects unique identifier used when logging into the system and the search topic. A tag description and an example can be found in appendix C.2.

[1] `http://www.itl.nist.gov/iaui/894.02/projects/trecvid/trecvid.tools/trec_eval_video/`

10 Conclusion and Future Work

"That nothing further remains to be done." [Roeder, 1993]

CARL FRIEDRICH GAUSS
German mathematician, astronomer and physicist (1777–1855)

Giving a final reflection on the finished work, this chapter draws a conclusion in summarising its cognitions and illustrates the course of the work in chapter 10.1. Chapter 10.2 summarises the findings of this thesis. In chapter 10.3, final remarks point to ideas and approaches that have not been considered in the developed software system but that are worth being focused on in a future work.

10.1 Conclusion

In this thesis, the development of a software tool for interactive video retrieval has been depicted. The structure has been oriented on a research procedure:
First of all, an introduction on relevant techniques and methods has been given. This includes a survey on video data processing, introduced in chapter 2. One video data format has been representatively presented (chapter 2.1) – MPEG-1 – which is the the data format of the videos used here. Following the design paradigm *"divide and conquer"*, it is useful to divide videos into smaller pieces. One common way has been introduced in chapter 2.2, shot boundary detection. For a successful retrieval, each divided part of the video (a "shot") needs a representative, e.g. for displaying in an interface. Chapter 2.3 presented the technique of automatic keyframe extraction for generating such a representative.
Chapter 3 delineated the problems and methods of resolution dealing with retrieval in video data. Currently, many projects such as the European K-Space [Izquierdo, 2005] are financed with the objective to find solutions for the problem presented in chapter 3.1, the semantic gap. Serious approaches to solve it were presented in the chapters 3.2, 3.3 and 3.4: the development of a semantic

visual feature ontology and the introduction of relevance feedback and based on that a query expansion.
The techniques presented so far were critically discussed in chapter 4.
Chapter 5 gave a survey on interactive video retrieval. The need of a video surrogate containing textual and visual information of a video were presented in chapter 5.1. A discussion on how to present video shots best in a retrieval system was hold in chapter 5.2. Chapter 5.3 presented two methods for video indexing: content-based and concept-based indexing. For creating a retrieval system that supports the user, it is important to find out more about his needs and preferences [Payette and Rieger, 1998]. Corresponding approaches were summarised in chapter 5.4. The survey ended with an introduction of two approaches: the Open Video Digital Library and the YouTube system (chapter 5.5). The commercial success of interactive video retrieval systems such as YouTube proves the demand for such services.
In chapter 6, the TRECVID workshop was introduced. It is the video track of the annual Text REtrieval Conference. The organiser NIST and its participants provide data for the research on video retrieval. The 2005 data set of TRECVID was presented in chapter 6.2. Developed approaches were introduced in chapter 6.3: The Informedia Digital Video Library from CMU, the Físchlár Digital Video System from DCU and the iBase system developed at ICL. Participants of the workshop can work on various given tasks. One of them, the search task, was presented in chapter 6.4. It is the relevant task for the development of the present system.
After that, the software engineering part of the thesis was documented in the chapters 7 (Design) and 8 (Implementation and Documentation). The engineering was oriented on the Object-Oriented Analysis and Design (OOAD) approach by Booch [1995].
Chapter 9 introduced two research ideas that could be realised using the developed system. It explained the need for another system for system evaluation and presented the results of a simulated user study. Finally, it introduced the conditions of the TRECVID user study.

10.2 Results of the study

The simulated test runs supported the hypothesis that “a combination system of implicit and explicit features is better than the system based on explicit feature only for video retrieval” A developed system S_1 including explicit relevance feedback returned less results than a system S_2 which considered both explicit and implicit relevance feedback.
To investigate the question “Which weighting factors for the different feature categories are appropriate for feedback?” three different interactive video retrieval models, S_2 - S_4, were implemented supporting textual and visual search queries. These models S_2 - S_4 consider both explicit and implicit relevance feedback using different feedback weighting methods. The result was that the

weighting factor used by model S_3 provided the best results. This model is based on considering the click on a keyframe (C1), the playing duration of a keyframe (C2) and other interactions with the keyframe (C3) with the weighting C1 > C2 > C3. The earlier the feedback is given the higher should be its weighting.

10.3 Future Work

As documented in the thesis, all requirements listed in chapter 7.1 have been fulfilled. However, the here developed system is everything but unmitigated! Considering the small time frame for a thesis like this, this realisation is not astonishingly. Similar systems as presented in chapter 6.3 reach much better retrieval results. Though they have been developed and improved over years with several researchers and programmers working on them. However, in the course of participating in the yearly TRECVID workshop, the framework for continuing research is given for the Information Retrieval Group at the University of Glasgow. The system can be used as a basis to implement, test and evaluate pursuing approaches (requirement (11)). One possible research direction is presented in chapter 9. As interactive video retrieval is a relatively new field of research, many improvements are imaginable. Moreover, comparing and determining similar systems means learning of their faults and their success. This chapter will give a short survey on approaches that should be considered in future work.

The system at hand can be used to perform interactive video retrievals. However, this is only *one* task in the field of video retrieval. TRECVID provides other areas that could be considered in progressing research (see chapter 6.1).
As proven by Campbell [2000b] (see also chapter 4.4), it is useful to implement more relevance feedback to the system for supporting the user in his search. The developed system mainly supports *explicit* relevance feedback. The hypothesis and the simulated user study in chapter 9 should be evaluated by running a real user test using the two developed systems.
Furthermore, the system only takes low-level features into account. High-level descriptors as presented in chapter 3.1 also provide useful information and should be considered.
The lightweight ontology presented in chapter 3.2 has been entirely ignored. A focus on this appears to be promising for improving interactive video retrieval systems.
The videos provided with the TRECVID data collection were recorded on a short period of time and the user experiments arrogated by NIST have to be executed within 15 minutes each, following default search topics. It would be interesting to see the efficiency of video retrieval systems in a daily operation with users running the system for retrieving topics *they* are interested in. There-

fore, it would be useful to create an up to date video data collection and to provide the system to participants for a more permanent usage to perform longer field studies. The researchers at DCU follow this approach with their Físchlár system which was introduced in chapter 6.3.2.
As the used computer language Java is platform independent, the developed tool can run similarly under diverse operational systems. It was tested on desktop machines running Microsoft Windows XP, Apple Macintosh and Linux. A challenge would be to adapt the interface to other devices such as handhelds. Foley et al. [2005] already conducted experiments using a tabletop device.

From the perspective of the interface, several improvements are conceivable:
Currently, it does not offer the opportunity to select between different visual features for comparison. The option to select between them should improve the retrieval. Sophisticated user interfaces offering this option were presented in chapter 6.
One improvement could be to investigate in finding the right moment when a system suggests new terms for query expansion. Currently, the user has to click a button to receive some suggestions. It is not said that this is the best solution to visualise this. Rather, users should have to click as less as possible.
The included Media Player from the Java Media Framework supports only a small number of video data formats. The official webpage of the API [1] have not been updated since late 2004. If the developer Sun has lost interest in further development of this framework, another solution has to be implemented to support other video formats.

[1] `http://java.sun.com/products/java-media/jmf/index.jsp`

A TREC Search Tasks

A.1 TRECVID 2004

The Search Tasks of TRECVID 2004 according to Smeaton et al. [2004]

125. Find shots of a street scene with multiple pedestrians in motion and multiple vehicles in motion somewhere in the shot.
126. Find shots of one or more buildings with flood waters around it/them.
127. Find shots of one or more people and one or more dogs walking together.
128. Find shots of U.S. Congressman Henry Hyde's face, whole or part, from any angle.
129. Find shots zooming in on the US capitol dome.
130. Find shots of a hockey rink with at least one of the nets fully visible from some point of view.
131. Find shots of fingers striking the keys on a keyboard which is at least partially visible.
132. Find shots of people moving a stretcher.
133. Find shots of Saddam Hussein.
134. Find shots of Boris Yeltsin.
135. Find shots of a person hitting a golf ball that then goes into the hole.
136. Find shots of Benjamin Netanyahu.
137. Find shots of one or people going up or down some visible steps or stairs.
138. Find shots of a handheld weapon firing.
139. Find shots of one or more bicycles rolling along.
140. Find shots of one or more umbrellas.

141. Find shots of Sam Donaldson's face – whole or part, from any angle, but including both eyes. No other people visible with him.
142. Find more shots of a tennis player contacting the ball with his or her tennis racket.
143. Find shots of one or more wheelchairs. They may be motorized or not.
144. Find shots of Bill Clinton speaking with at least part of the US flag visible behind him.
145. Find shots of one or more horses in motion.
146. Find shots of one or more skiers skiing a slalom course with at least one gate pole visible.
147. Find shots of one or more buildings on fire, with flames and smoke visible.
148. Find shots of one or more signs or banners carried by people at a march or protest.

A.2 TRECVID 2005

The Search Tasks of TRECVID 2005 according to Smeaton and Ianeva [2005]

149. Find shots of Condoleeza Rice.
150. Find shots of Iyad Allawi, the former prime minister of Iraq.
151. Find shots of Omar Karami, the former prime minister of Lebannon.
152. Find shots of Hu Jintao, president of the People's Republic of China.
153. Find shots of Tony Blair.
154. Find shots of Mahmoud Abbas, also known as Abu Mazen, prime minister of the Palestinian Authority.
155. Find shots of a graphic map of Iraq, location of Baghdad marked – not a weather map.
156. Find shots of tennis players on the court – both players visible at the same time.
157. Find shots of people shaking hands.
158. Find shots of a helicopter in flight.
159. Find shots of George Bush entering or leaving a vehicle (e.g., car, van, airplane, helicopter, etc.), he and vehicle both visible at the same time.
160. Find shots of something (e.g., vehicle, aircraft, building, etc.) on fire with flames and smoke visible.

161. Find shots of people with banners or signs.
162. Find shots of one or more people entering or leaving a building.
163. Find shots of a meeting with a large table and more than two people.
164. Find shots of a ship or boat.
165. Find shots of basketball players on the court.
166. Find shots of one or more palm trees.
167. Find shots of an airplane taking off.
168. Find shots of a road with one or more cars.
169. Find shots of one or more tanks or other military vehicles.
170. Find shots of tall building (with more than 5 floors above the ground).
171. Find shots of a goal being made in a soccer match.
172. Find shots of an office setting, i.e., one or more desks/tables and one or more computers and one or more people.

A.2.1 Topic Types

	Named			Generic		
Topic	**Person**	**Event**	**Place**	**Person**	**Event**	**Place**
149	×					
150	×					
151	×					
152	×					
153	×					
154	×					
155				×		
156				×		×
157				×	×	
158				×	×	
159	×			×	×	
160				×	×	
161				×		
162				×	×	
163				×	×	
164				×		
165				×		×
166				×		
167				×	×	
168				×		×
169				×		
170				×		
171					×	
172				×		×

Table A.1: 2005 Topic types [Over et al., 2005]

B Interface Proposals

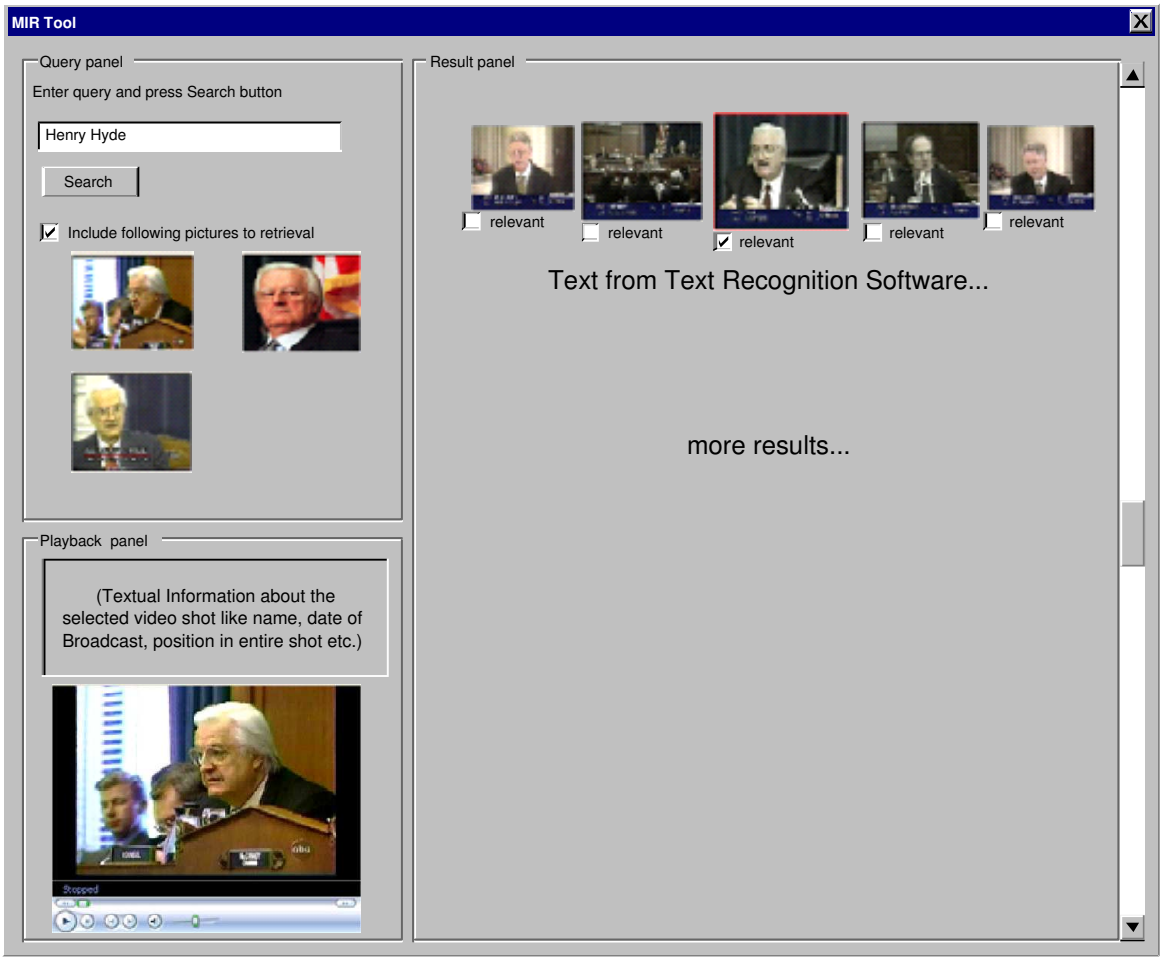

Figure B.1: Proposed Interface 1

Interface proposal 1 is mainly oriented on the Físchlár system from DCU as it is one of the most matured systems available. It is divided into three main panels: query panel, result panel and playback panel. The query panel includes textual queries and visual queries either from example images or from other frames previously declared as relevant by the user. The result panel shows the retrieved results in its context which means, that the main keyframe is surrounded by its neighboured frames. Relevant keyframes can be added to the next search by activating the check box

under every frame. Supplementary, the text from the Text Recognition Software is displayed. The playback panel gives additional information about the shot like the title, time and date of broadcast, the position of the shot in the entire video et cetera. It uses the Microsoft Media Player for playback.

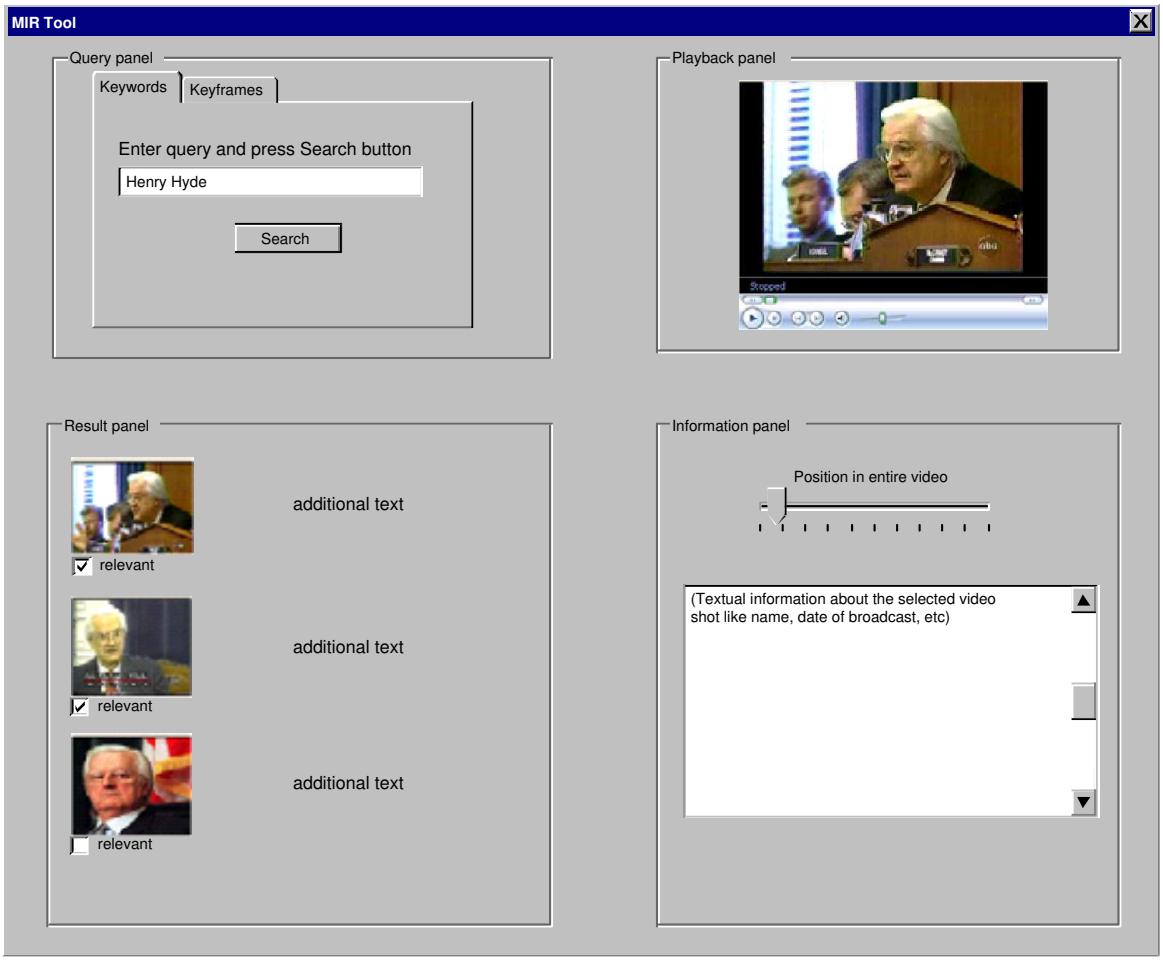

Figure B.2: Proposed Interface 2

The second interface proposal is divided into four parts: query panel, result panel, playback panel and information panel. The query panel consists of two registers: The keyword register and the keyframe register. In figure B.2, the keyword register is activated. Here, the user can only enter a textual search query. The second register gives him the chance to add example images. The result panel lists the retrieved results, The user can click a check box under every keyframe, if it is a relevant keyframe for his search. Selecting an image automatically switches to the keyframe register in the search panel. The image will be added to the next search query. Additional text in the result panel gives more information about the listed video shot. The shot can be payed using

the Microsoft Media Player in the playback panel. Additional information can be found then in the information panel.

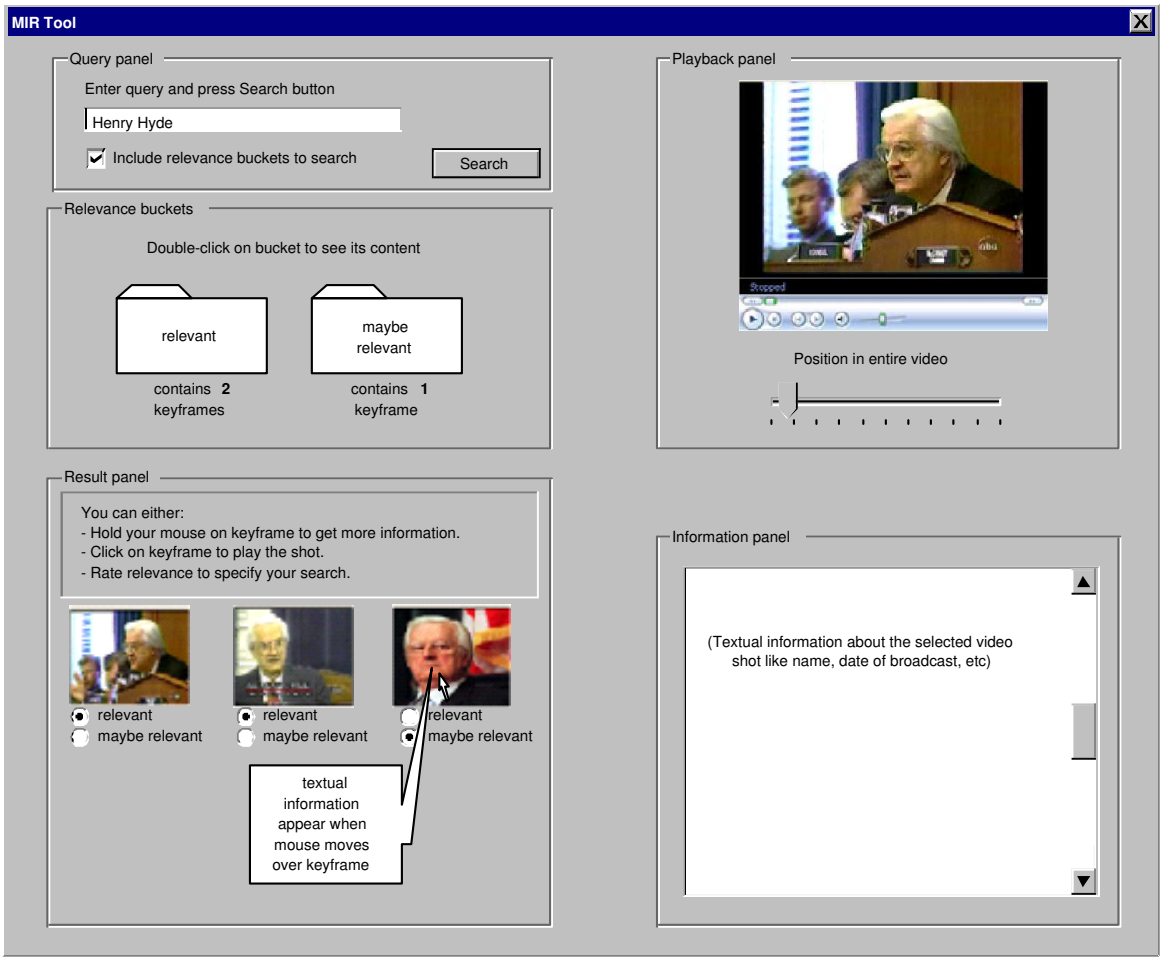

Figure B.3: Proposed Interface 3

Proposal B.3 is divided into five panels: A little search panel, the result panel, relevance buckets, the playback panel and the information panel. In the search panel, the user can enter a query. After pressing the search button, he gets some results in the result panel. The resulting keyframes are listed in order of their relevance. The user can hold the mouse button over a keyframe to display more textual information about the shot using the tool tip technique. Moreover, he can play the video shot in the playback panel by clicking on the keyframe. In the result panel, he has also the opportunity to rate the listed images as *relevant* or *maybe relevant* using radio buttons. Rated keyframes are stored in the accordant bucket in the relevance buckets panel. The user can view and change the content of the buckets by double-clicking the bucket (can not be seen on the figure). These rated keyframes can be added to the next search by activating the *Include relevance buckets to search* option in the search panel. As mentioned before, the playback panel plays the selected

video shot. It also displays the position of the shot in the entire broadcast. The information panel contains more information about the selected video.

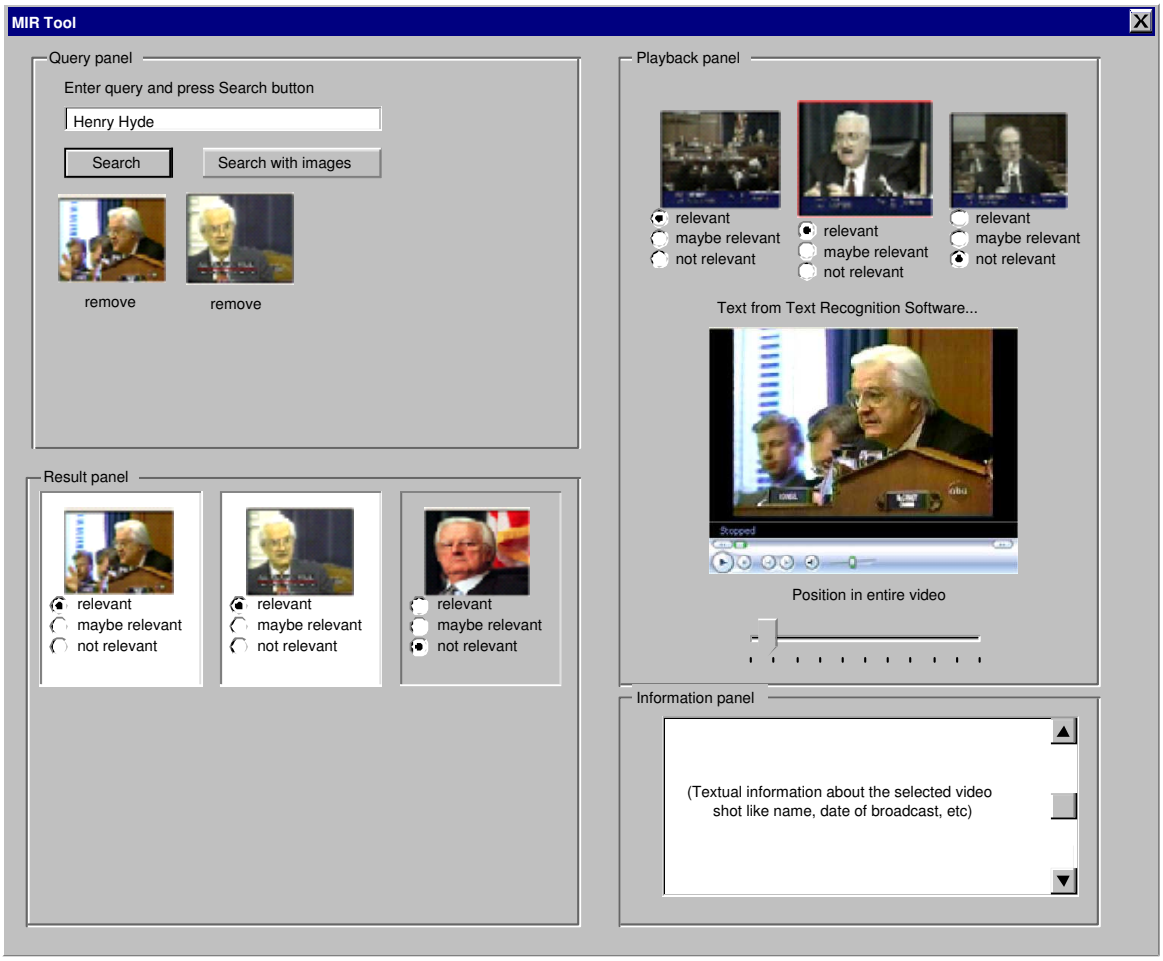

Figure B.4: Proposed Interface 4

Proposal B.4 consists of four panels: The query panel, the result panel, the playback panel and the information panel. The query panel contains a text box for the textual query but also includes keyframes which are declared relevant by the user in preceding searches. Depending on the button he uses, the user can either trigger a new search with or without including these keyframes. The retrieved images are listed in the result panel. Here, the user can rate them as relevant, maybe relevant or not relevant. In the proposal, this is realised using radio buttons. A representation using icons is also conceivable. Keyframes which have been listed in a precedent search are highlighted, so the user can easily differentiate between new and old results. After clicking on a keyframe, the video starts playing in the playback panel. A bar shows the position of the shot in the entire broadcast. Furthermore, the keyframe is displayed in its context in showing its neighboured keyframes.

Here, it is also possible to mark the relevance of a keyframe. Textual information are presented in the information panel.

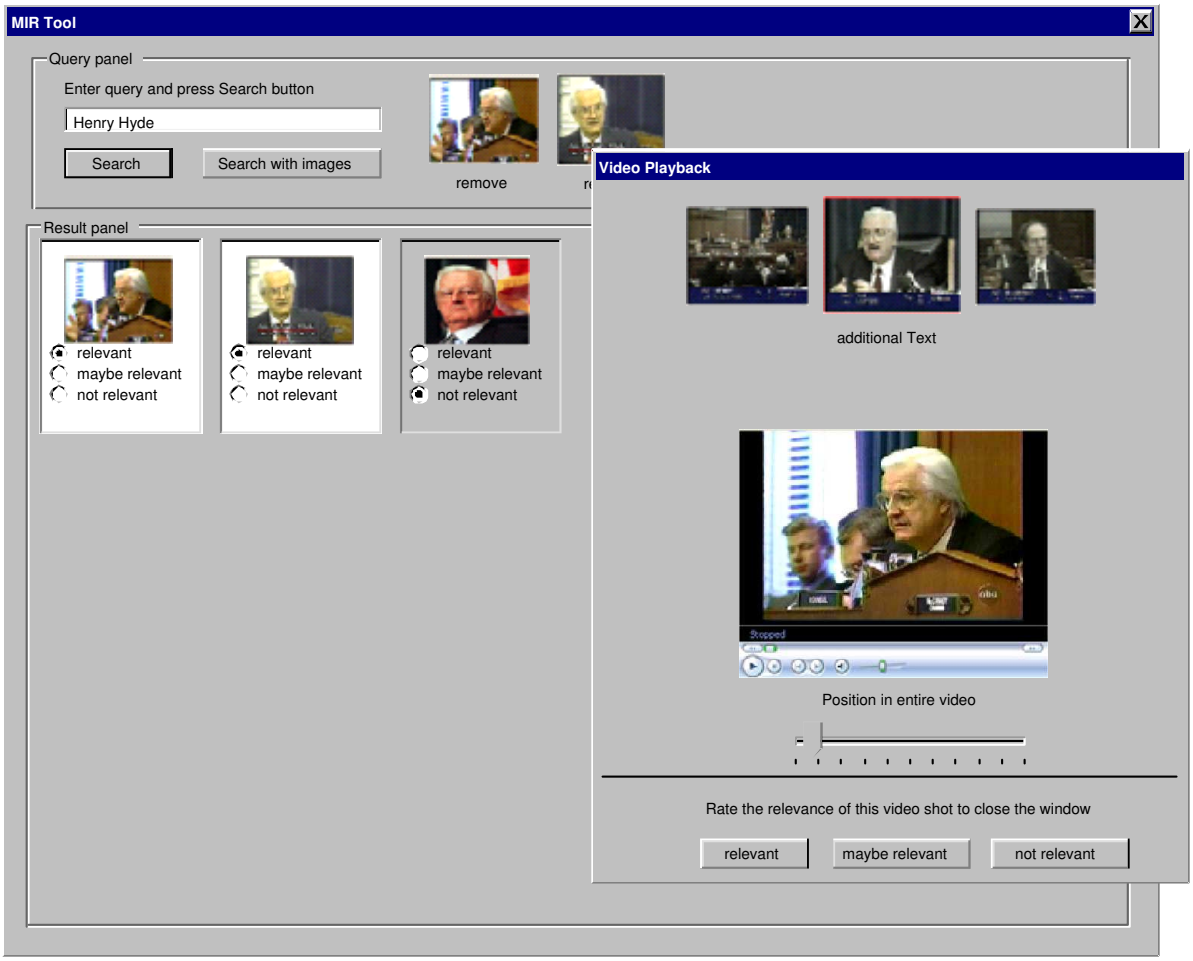

Figure B.5: Proposed Interface 5

Proposal B.5 is similar to proposal B.4. It is divided into the search panel and the result panel which have the same features as their equivalents in proposal B.4. Whether the user wants to play a video, a new window opens. It contains the keyframe and its neighboured frames, some textual information, the player itself, the position of the shot in the entire video and different buttons for rating the relevance of the retrieved video. As the window has no close button, the only chance to close the window is in pressing one of the rating buttons. This is a solution to force the user to give a relevance feedback, as he else wise might be to convenient to do so. Another advantage is the possibility to change the size of the window including the size of the video.

Figure B.6: Proposed Interface 6

Proposal B.6 combines ideas from the preceding proposals. It contains a query panel consisting of a text box for the initial query, a search button and also three relevance buckets: *relevant, maybe relevant* and *not relevant*. After triggering a new search pressing the button, the retrieved results are presented in the result panel. Here, the user can rate their relevance. Keyframes which have been listed in a precedent search are highlighted, so the user can easily differentiate between new and old results.

Whether the user wants to play a video, a new window opens. It contains the keyframe and its neighboured frames, some textual information, the player itself, the position of the shot in the entire video and different buttons for rating the relevance of the retrieved video. As the window has no close button, the only chance to close the window is in pressing one of the rating buttons. This is a solution to force the user to give a relevance feedback, as he else wise might be to convenient to do so. Another advantage is the possibility to change the size of the window including the size of the video.

Depending on rating, the user can store or buffer relevant keyframes. When clicking on a bucket

in the search panel, it is highlighted in another colour and its content is displayed on the right hand side of the interface which is marked in the same colour as the selected bucket. Here, the user can rate them again to move them into other buckets.

Figure B.7: Proposed Interface 7

Figure B.7 shows the last proposal. It unites the best ideas of it's prior proposals and strikes a balance between all of them. After entering a query in the search panel, results get listed in the result panel. Here, the user can rate the relevance of the retrieved results as *relevant, maybe relevant* and *not relevant*. Keyframes which have been listed in a precedent search are highlighted, so the user can easily differentiate between new and old results. The appearance of the relevance buckets which was first introduced in proposal B.3 has changed. Now, they are positioned under the result panel in another layer. The user can switch between these layers using tabs. Empty tabs are disabled.

The playback panel contains the keyframe and its neighboured frames, some textual information,

the player itself, the position of the shot in the entire video and different buttons for rating the relevance of the retrieved video.

C Logging files

C.1 Video Search Result Log

C.1.1 Log files for submission

Example of a log file[1] listing the shots a user retrieved for a search topic.

```
<!−− Example video search results −−>
<!DOCTYPE videoSearchResults SYSTEM "videoSearchResults.dtd">
<videoSearchResults>
    <videoSearchRunResult pType="I" trType="C" sysId="SiriusCy1" priority="1" condition="1"
    desc="This␣interactive␣run␣uses␣local␣ASR␣and␣all␣the␣features␣donated␣by␣DCU">
        <videoSearchTopicResult tNum="075" elapsedTime="9.3" searcherId="A">
            <item seqNum="1" shotId="shot118_2"/>
            <item seqNum="2" shotId="shot118_3"/>
            <item seqNum="3" shotId="shot18_19"/>
            <item seqNum="4" shotId="shot123_2"/>
            <item seqNum="5" shotId="shot56_42"/>
            <item seqNum="6" shotId="shot193_3"/>
            <item seqNum="7" shotId="shot121_12"/>
            <item seqNum="8" shotId="shot22_20"/>
            <item seqNum="9" shotId="shot103_122"/>
            <!−− ... −−>
            <item seqNum="1000" shotId="shot118_2"/>
        </videoSearchTopicResult>
        <!−− ... −−>
        <videoSearchTopicResult tNum="099" elapsedTime="2.9" searcherId="Z">
            <item seqNum="1" shotId="shot118_2"/>
            <item seqNum="2" shotId="shot118_3"/>
            <item seqNum="3" shotId="shot18_19"/>
```

[1] taken from `http://www-nlpir.nist.gov/projects/tv2005/dtds/videoSearchResults.xml`

```
                    <item seqNum="4" shotId="shot123_2"/>
                    <item seqNum="5" shotId="shot56_42"/>
                    <item seqNum="6" shotId="shot193_3"/>
                    <item seqNum="7" shotId="shot121_12"/>
                    <item seqNum="8" shotId="shot22_20"/>
                    <item seqNum="9" shotId="shot103_122"/>
                    <!-- ... -->
                    <item seqNum="1000" shotId="shot118_2" />
                </videoSearchTopicResult>
            </videoSearchRunResult>
</videoSearchResults>
```

C.1.2 Log files for internal evaluation

Example of a log file created for evaluation. It contains all retrieved and rated shots arranged by a weighting of their relevance.

```
0149 0 shot61_30 1 999.0 4711
0149 0 shot65_47 2 199.0 4711
0149 0 shot11_178 3 198.0 4711
0149 0 shot19_44 4 197.0 4711
0149 0 shot10_35 5 196.0 4711
0149 0 shot8_154 6 195.0 4711
0149 0 shot5_191 7 194.0 4711
0149 0 shot53_455 8 193.0 4711
0149 0 shot37_21 9 192.0 4711
0149 0 shot108_164 10 191.0 4711
0149 0 shot24_39 11 190.0 4711
0149 0 shot83_52 12 189.0 4711
0149 0 shot35_151 13 188.0 4711
0149 0 shot116_251 14 187.0 4711
0149 0 shot70_127 15 186.0 4711
0149 0 shot100_32 16 185.0 4711
0149 0 shot24_40 17 184.0 4711
0149 0 shot108_249 18 183.0 4711
0149 0 shot37_24 19 182.0 4711
```

C.2 User Behaviour Log

C.2.1 Tag description

The tags used in the behaviour logs are described in the tables below.

Tag	Meaning
USER	User ID
TOPIC	Topic ID
RUNID	Run ID

Table C.1: general information

Tag	Meaning
CLICKADDTERM	New term field
CLICKCANCEL	Cancel button
CLICKEXPAND	Expand query button
CLICKERASE	Erase results button
CONFIRMERASE	Confirm erase
CLICKSTARTTOPIC	Start new topic button

Table C.2: general interaction tags

Tag	Meaning
VQCANDIDATE	Visual query candidate
EXPTERM	term from query expansion
VQUERYSIZE	visual query size
TQUERY	textual query
ETQUERY	textual query after expansion

Table C.3: query expansion tags

Tag	Meaning
PASTE	paste query (from Playback Panel)
CUTQ	cut query (Search Panel)
PASTEQ	paste query (Search Panel)
ADD2VQ	add to visual query list
REMFROMVQ	remove from visual query list
DDATQUERY	disable date query
EDATQUERY	enable date query

Table C.4: queries and query modification tags

Tag	Meaning
CLICK	click on keyframe
RATER	rate relevant
RATEMR	rate maybe relevant
RATEIR	rate not relevant
RATECKR	rate calculated keyframe relevant
RATECKMR	rate calculated keyframe maybe relevant
RATECKIR	rate calculated keyframe not relevant
RATECKFR	rate calculated keyframe as final result
BROWSEL	click on left neighboured keyframe
BROWSER	click on right neighboured keyframe

Table C.5: retrieval strategy (action) tags

C.2.2 Example log

Tue Aug 08 12:56:15 BST 2006
INFO Tue Aug 08 12:56:15 BST 2006 USER: Frank
INFO Tue Aug 08 12:56:15 BST 2006 TOPIC: 0149
INFO Tue Aug 08 12:56:15 BST 2006 RUNID: 4711
INFO Tue Aug 08 12:56:20 BST 2006 TQUERY: bush
INFO Tue Aug 08 12:56:28 BST 2006 CLICK: /collection/TRECVID2005_132/shot132_5_RKF.jpg
INFO Tue Aug 08 12:56:33 BST 2006 RATECKIR: /collection/TRECVID2005_132/shot132_5_RKF.jpg
INFO Tue Aug 08 12:56:39 BST 2006 CLICK: /collection/TRECVID2005_61/shot61_30_RKF.jpg
INFO Tue Aug 08 12:56:42 BST 2006 RATECKFR: /collection/TRECVID2005_61/shot61_30_RKF.jpg
INFO Tue Aug 08 12:56:45 BST 2006 CLICKEXPAND
INFO Tue Aug 08 12:56:45 BST 2006 VQCANDIDATE: /collection/TRECVID2005_132/shot132_5_RKF.jpg
INFO Tue Aug 08 12:56:45 BST 2006 VQCANDIDATE: /collection/TRECVID2005_61/shot61_30_RKF.jpg
INFO Tue Aug 08 12:56:45 BST 2006 EXPTERM: plai
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: bush
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: call

INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: secur
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: watch
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: secretari
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: think
INFO Tue Aug 08 12:56:46 BST 2006 EXPTERM: help
INFO Tue Aug 08 12:56:50 BST 2006 ADD2VQ /collection/TRECVID2005_132/shot132_5_RKF.jpg
INFO Tue Aug 08 12:57:00 BST 2006 ETQUERY: +bush +usa
INFO Tue Aug 08 12:57:00 BST 2006 VQUERYSIZE 1

Bibliography

A. Abramson and H. Walitsch. *Die Geschichte des Fernsehens*. Fink, Paderborn, Germany, 01 2003.

N. Adami, R. Leonardi, and P. Migliorati. An Overview of Multi-modal Techniques for the Characterization of Sport Programmes. In *Proc. Conf. Visual Communications and Image Processing, Lugano, Switzerland*, pages 1296–1306, 7 2003.

P. Aigrain, H. Zhang, and D. Petkovic. Content-based representation and retrieval of visual media: A state-of-the-art review. *Multimedia Tools and Applications*, 3:179–202, 1996.

Alexa Internet. Top Sites. `http://www.alexa.com/site/ds/top_sites?ts_mode=global&lang=none`, 2006. last checked: 23.06.2006.

G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

L. H. Armitage and P. G. B. Enser. Information Need in the Visual Document Domain. Technical report, School of Information Management, Universitz of Brighton, 2005. Report on Project RDD/G/235 to the British Library Research and Innovation Centre.

R. Attar and A. S. Fraenkel. Local Feedback in Full-Text Retrieval Systems. *Journal of the ACM*, 24(3):397–414, 1977.

W. Bailer, F. Höller, A. Messina, D. Airola, P. Schallauer, and M. Hausenblas. State of the Art of Content Analysis Tools for Video, Audio and Speech. Technical report, PrestoSpace, 3 2005. Deliverable D15.3 MDS3.

M. Beaulieu. Experiments with interfaces to support query expansion. *Journal of Documentation*, 53(1):8–19, 1997.

R. K. Belew. *"Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW"*. Cambridge University Press, 2000.

G. Booch. *Objektorientierte Analyse und Design*. Addison-Wesley, 01 1995.

P. Browne, C. Czirjek, C. Gurrin, R. Jarina, H. Lee, S. Marlow, K. McDonald, N. Murphy, N. O'Connor, A. F. Smeaton, and J. Ye. Dublin City University Video Track Experiments for TREC 2002. In *TREC2002 – Text REtrieval Conference, Gaithersburg, Maryland, 19-22 November 2002*, 2002.

P. Browne and A. F. Smeaton. Video Retrieval Using Dialogue, Keyframe Similarity and Video Objects. In *ICIP 2005 – International Conference on Image Processing, Genova, Italy*, 9 2005.

P. Browne, A. F. Smeaton, N. Murphy, N. O'Connor, S. Marlow, and C. Berrut. Evaluating and Combining Digital Video Shot Boundary Detection Algorithms. In *IMVIP 2000 – Irish Machine Vision and Image Processing Conference. Belfast, Northern Ireland*, 2000.

I. Campbell. Interactive evaluation of the Ostensive Model, using a new test-collection of images with multiple relevance assessments. In *Journal of Information Retrieval*, volume 1, pages 89–114, 2000a.

I. Campbell. *The ostensive model of developing information needs*. PhD thesis, University of Glasgow, UK, 2000b.

I. Campbell and C. J. van Rijsbergen. The ostensive model of developing information needs. In *Proc. of CoLIS-96, 2nd Int. Conf. on Conceptions of Library Science*, pages 251–268, 1996.

Carnegie Mellon University. Informedia – Digital Video Library System. Annual Progress Report, 02 1998.

Carnegie Mellon University. The Informedia Project – Research Timeline. `http://www.informedia.cs.cmu.edu/timeline/index.html`, 2006. last checked: 24.02.2006.

C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

Centre for Digital Video Processing. The Físchlár System. Fact Sheet, 06 2005.

D. C. Chmielewski and C. Gaither. Google Goes From Web to Webster's. *Los Angeles Times*, pages 397–414, 7 2006.

M. Christel and R. Concescu. Addressing the Challenge of Visual Information Access from Digital Image and Video Libraries. In *Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (Denver, CO, June 2005)*, pages 69–78, 2005.

M. Christel, A. Hauptmann, A. S. Warmack, and S. A. Crosby. Adjustable filmstrips and skims as abstractions for a digital video library. In *Proceedings of IEEE forum on Research and Technology Advances in Digital Libraries*, Baltimore, Maryland, USA, 05 1999.

M. Christel, J. Yang, R. Yan, and A. Hauptmann. Carnegie Mellon University Search. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

M. Claypool, P. Le, M. Waseda, and D. Brown. Implicit interest indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, pages 33–40, 2001.

CollabNet. bindMark Project. `https://bindmark.dev.java.net/`, 2005. last checked: 25.05.2006.

E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. J. F. Jones, H. Le Borgne, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. E. O'Connor, N. O'Hare, S. Rothwell, A. F. Smeaton, and P. Wilkins. TRECVID 2004 Experiments in Dublin City University. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

J. Cox, L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval retrieval system, PicHunter: Theory, implementation and psychophysical experiments. In *IEEE Trans. Image Processing*, volume 9, pages 20–37, 1 2000.

P. J. Denning. The Locality Principle. *Communication of the ACM*, 48(7):19–24, 2005.

E. W. Dijkstra. The Humble Programmer. *Communication of the ACM*, 15(10):859–866, 1972.

N. Dimitrova, T. McGee, and H. Elenbaas. Video keyframe extraction and filtering: A keyframe is not a keyframe to everyone. In F. Golshani and K. Makki, editors, *Proceedings of the Sixth International Conference on Information and Knowledge Management (CIKM'97), Las Vegas, Nevada, November 10-14, 1997*, pages 113–120. ACM, 1997.

W. Ding, G. Marchionini, and D. Soergel. Multimodal surrogates for video browsing. In *Proceedings of the Fourth ACM conference on Digital Libraries*, Berkeley, CA, USA, 08 1999.

E. N. Efthimiadis. Query Expansion. *Annual Review of Information Science and Technology*, 31, 1996.

P. Enser. Visual image retrieval: seeking the alliance of concept-based and content-based paradigms. *Journal of Information Science*, 26(4):199–210, 08 2000.

P. Enser and C. Sandom. Retrieval of archival moving imagery - cbir outside the frame? In *CIVR '02: Proceedings of the International Conference on Image and Video Retrieval*, pages 206–214, London, UK, 2002. Springer-Verlag.

W. Faulstich. *Filmgeschichte*. Fink, Paderborn, Germany, 03 2005.

E. Foley, C. Gurrin, G. Jones, C. Gurrin, G. Jones, H. Lee, S. McGivney, N. E. O'Connor, S. Sav, A. F. Smeaton, and P. Wilkins. TRECVid 2005 Experiments at Dublin City University. In *TRECVid 2005 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Maryland, 14-15 November 2005*, 2005.

H. Fowkes and M. Beaulieu. Interactive searching behaviour: Okapi experiment for TREC-8. In *Proceedings of IRSG 2000. 22nd Annual Colloquium on Information Retrieval Research*, 2002.

G. Geisler. The Open Video Project: Redesigning a Digital Video Digital Library. Presentation at the American Society for Information Science and Technology Information Architecture Summit, 02 2004. Austin, Texas, USA.

A. Girgensohn, J. Adcock, M. Cooper, and L. Wilcox. A Synergistic Approach to Efficient Interactive Video Retrieval. In *Proc. Human-Computer Interaction INTERACT 2005, LNCS 3585*, pages 781–794. Technische Hogeschool Eindhoven, The Netherlands, 2005.

Y. Gong, C. H. Chuan, and G. Xiaoyi. Image indexing and retrieval based on colour histograms. *Multimedia Tools and Applications*, 2:133–156, 1996.

A. A. Goodrum. Multidemensional scaling of video surrogates. *Journal of the American Society for Information Science*, 52(2):174–182, 2001.

H. Greisdorf and B. O'Connor. Modelling what users see when they look at images: a cognitive viewpoint. *Journal of Documentation*, 58(1):6–29, 2002.

Grundig Intermedia GmbH. Press Release: Der Markt für DVD-Geräte wird weiter wachsen. `http://www.grundig.de/inc/presse.grundig/produktmeldungen/Video__DVD/De%r_DVD_Markt__waechst/index.html`, 05 2005. last checked: 12.02.2006.

C. Gurrin, D. Johansen, and A. F. Smeaton. *Supporting Relevance Feedback in Video Search*, pages 561–564. Springer, Berlin / Heidelberg, Germany, 2006.

G. Haberfehlner. Development of a System for Automatic Dialog Scene Detection. Master's thesis, Fachhochschule Hagenberg, Austria, 2004.

A. Hauptmann. [Trecvid] RE: MS ASR output format guesses. eMail to the mailing list `trecvid2005@nist.gov`, 06 2005.

A. Hauptmann, M. Y. Chen, M. Christel, C. Huang, W. Lin, T. Ng, N. Papernick, A. Velivelli, J. Yang, R. Yan, H. Yang, and H. D. Wactlar. Confounded Expectations: Informedia at TRECVID 2004. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

A. Hauptmann, M. Christel, R. Concescu, J. Gao, Q. Jin, W. Lin, J. Y. Pan, S. M. Stevens, R. Yan, J. Yang, and Y. Zhang. CMU Informedia's TRECVID 2005 Skirmishes. In *TRECVid 2005 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Maryland, 14-15 November 2005*, 2005.

L. He, E. Saocki, A. Gupta, and J. Grudin. Auto-Summarization of Audio-Video Presentations. In *ACM Multimedia 99*, 1999.

D. Heesch, P. Howarth, J. Magalhães, A. May, M. Pickering, A. Yavlinski, and S. Rüger. Video Retrieval using Search and Browsing. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

C. A. R. Hoare. The Emperor's old clothes. *Communications of the ACM*, 24(2):75–83, 2 1981.

A. Hughes, T. Wilkens, B. Wildemuth, and G. Marchionini. Text or Pictures? An Eyetracking Study of How People View Digital Video Surrogates. *Image and Video Retrieval: Second International Conference*, 2727:271–280, 2003.

Human/Computer Interaction Laboratory. Online Survey Design Guide. `http://lap.umd.edu/survey_design/defs.html`, 2006. last checked: 30.07.2006.

Imperial College London. Multimedia and Information Systems: Research. `http://mmis.doc.ic.ac.uk/research.html`, 2006. last checked: 27.02.2006.

Information Retrieval Goup. Terrier – TERabyte RetrIEveR. `http://ir.dcs.gla.ac.uk/terrier/`, 03 2005. last checked: 19.05.2006.

L. Ingber. Statistical mechanics of neocortical interactions: Stability and duration of the 7+-2 rule of short-term-memory capacity. *Physical Review A*, 31(2):1183–1186, 2 1985.

Interaction Design Laboratory. The Open Video Project. `http://www.open-video.org/index.php`, 2006. last checked: 20.06.2006.

E. Izquierdo. Knowledge Space of Shared Technology and Integrative Research to Bridge the Semantic Gap. Technical report, Network of Excellence, 3 2005. Strategic Objective 2.4.7, "Semantic-based Knowledge and Content Systems", FP6-2004-IST, 4th Call.

G. Jefferson. Video websites pop up, invite postings. *USA Today*, 11 2005.

R. Jesus, J. Magalhães, A. Yavlinski, and S. Rüger. Imperial College at TRECVID. In *TRECVid 2005 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Maryland, 14-15 November 2005*, 2005.

J. Johnson Lewis. Joan Baez Quotes. About Women's History. `http://womenshistory.about.com/od/quotes/a/joan_baez.htm`, 1997. last checked: 21.08.2006.

J. Joubert and P. Auster. *The Notebooks of Joseph Joubert*. The New York Review of Books, New York, USA, 1983.

D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–384, New York, NY, USA, 2004. ACM Press.

V. Kobla, D. Doermann, and C. Faloutsos. Developing High-Level Representations of Video Clips using VideoTrails. In *JProceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases VI*, pages 81–92, 1998.

J. Koenemann and N. J. Belkin. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of the Human Factors in Computing Systems Conference (CHI'96)*, pages 205–212, 1996.

M. Koskela, A. F. Smeaton, and G. Gaughan. Semantic Analysis of Concept Models for News Video. In V. Bonnardel, M. Oakes, and John Tait, editors, *Workshop: Visual Categorisation and Image Management Systems*, Sunderland, United Kingdom, 06 2006. University of Sunderland.

L. R. Lieberman. Words versus objects: comparison of free verbal recall. *Psychol. Rep.*, 17: 983–988, 1965.

R. Lienhart, S. Pfeiffer, and W. Effeisberg. Video abstracting. *Communications of the ACM*, 40 (12):55–62, 1997.

C. A. Lindley. A multiple-interpretation framework for modeling video semantics. In *ER-97 Workshop on Conceptual Modeling in Multimedia Information Seeking*, 1997.

H. D. Luckhardt. Virtuelles Handbuch Informationswissenschaft. `http://is.uni-sb.de/studium/handbuch/exkurs.ir.html`, 07 2006. last checked: 23.07.2006.

A. Maass and A. Russo. Directional Bias in the mental Represenation of spatial events: Nature or Culture? *Psychological Science*, 14(4):296–301, 07 2003.

M. Magennis and C. J. van Rijsbergen. The potential and actual effectiveness of interactive query expansion. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–332, New York, NY, USA, 1997. ACM Press.

J. Mandler and G. H. Ritchey. Long term memory for pictures. *Journal of Experimental Psychology*, 3:386–396, 1977.

G. Marchionini. *Information seeking in electronic environments*. Cambridge University Press, 1995.

G. Marchionini. Augmenting library services: Toward the sharium. In *Proceedings of International Symposium on Digital Libraries 1999*, pages 40–47, 09 1999.

G. Marchionini. Video and Learning Redux: New Capabilities for Practical Use. *Educational Technology*, 03 2003.

G. Marchionini and G. Geisler. The Open Video Digital Library. *D-Lib Magazine*, 8(12), 12 2002.

G. Marchionini, V. Nolet, H. Williams, W. Ding, J. Beale, A. Rose, A. Gordon, E. Enomoto, and L. Harbinson. Content + Connectivity => Community: Digital Resources for a Learning Community. In *Proceeding of ACM Digital Libraries 97*, pages 212–220, 07 1997.

M. Markkula and E. Sormumen. Searching for photos: journalist's practices in pictorial IR. In *The challenge of Image Retrieval Research Workshop*, Newcastle upon Tyne, United Kingdom, 1998.

J. Mayer and J. P. Holms. *Bite-size Einstein: Quotations on just about everything from the Greatest Mind of the twentieth century*. St. Martin's Press, New York, USA, 1996.

S. Means and M. Bodie. *"The Book of SAX: The Simple API for XML"*. No Starch Press, 06 2002.

R. K. Merton. *On the Shoulders of Giants*. The University of Chicago Press, Chicago, USA, 1993.

V. Mezaris, H. Doulaverakis, S. Herrmann, B. Lehane, N. O'Connor, I. Kompatsiaris, and M. G. Strintzis. Combining textual and visual information processing for interactive video retrieval: SCHEMA's participation to TRECVID 2004. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

X. Mu. Content-based Video Retrieval: Does Video's Semantic Visual Feature Matter? Milwaukee, USA, 2006. University of Wisconsin-Milwaukee.

X. Mu and G. Marchionini. Enriched Video Semantic Metadata: Authorization, Integration, and Presentation. In *ASIST 2003 Annual Meeting – Humanizing Information Technology: From Ideas to Bits and Back*, Westin Long Beach, California, USA, 10 2003a.

X. Mu and G. Marchionini. Statistical visual feature indexes in video retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 395–396, New York, NY, USA, 2003b. ACM Press.

P. Munesawang and L. Guan. Adaptive Video Indexing and Automatic/Semi-Automatic Relevance Feedback. *IEEE Transactions on circuits and systems for video technology*, 15(8):1032–1046, 8 2005.

J. Naisbitt. *Megatrends*. Warner Books, 1982.

M. R. Naphade, L. Kennedy, J. R. Kender, S. Chang, J. R. Smith, P. Over, and A. Hauptmann. A Light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005, 2005.

M. R. Naphades, R. R. Wang, and T. S. Huang. Audio-visual query and retrieval: A system that uses dynamic programming and relevance feedback. *Journal of Electronic Imaging*, 10(4):861–870, 10 2001.

NIST. Two designs for interactive video search experiments. `http://www-nlpir.nist.gov/projects/tv2003/TRECVID_Interactive_Experimen%tal_Design.html`, 2003. last checked: 01.06.2006.

NIST. Guidelines for the TRECVID 2003 Evaluation. `http://www-nlpir.nist.gov/projects/tv2003/tv2003.html`, 06 2004a. last checked: 14.02.2006.

NIST. TREC: Overview. `http://trec.nist.gov/overview.html`, 01 2004b. last checked: 14.02.2006.

NIST. Guidelines for the TRECVID 2004 Evaluation. `http://www-nlpir.nist.gov/projects/tv2004/tv2004.html`, 02 2005. last checked: 23.02.2006.

NIST. Call for participation in TRECVID 2006. `http://www-nlpir.nist.gov/projects/tv2006/call.html`, 02 2006a. last checked: 14.02.2006.

NIST. Guidelines for the TRECVID 2005 Evaluation. `http://www-nlpir.nist.gov/projects/tv2005/tv2005.html`, 01 2006b. last checked: 23.02.2006.

NIST. Guidelines for the TRECVID 2006 Evaluation. `http://www-nlpir.nist.gov/projects/tv2006/tv2006.html`, 02 2006c. last checked: 22.02.2006.

NIST. List of the participanting groups and the tasks they have chosen. `http://www-nlpir.nist.gov/projects/trecvid/tv6.participants`, 02 2006d. last checked: 27.03.2006.

NIST. TRECVID Data Availability. `http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html`, 02 2006e. last checked: 17.05.2006.

Object Management Group. Unified Modeling Language Specification Version 2.0. `http://www.omg.org/technology/documents/formal/uml.htm`, 01 2006. last checked: 30.03.2006.

C. O'Toole. Evaluation of Shot Boundary Detection on a Large Video Test Suite. In *Proceedings of Challenges in Image Retrieval*, Newcastle, UK, 02 1999.

I. Ounis. Terrier – A practical overview. Technical report, Information Retrieval Group, University of Glasgow, United Kingdom, 12 2005. Slides.

I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR 05), Santiago de Compostela, Spain*, 2005.

P. Over. [Trecvid] IMPORTANT: Additional/replacement ASR.MT and Status. eMail to the mailing list `trecvid2005@nist.gov`, 06 2005.

P. Over, I. Tzaveta, W. Kraaij, and A. F. Smeaton. Trecvid 2005 – an overview. In *TREC video retrieval evaluation online proceedings*, 2005.

D. L. Parnas and J. Madey. Functional documentation for computer systems. *Science of Computer Programming*, 25(1):41–61, 1995.

S. D. Payette and O.Y. Rieger. Supporting scholarly inquiry: Incorporating users in the design of the digital library. *Journal of Academic Librarianship*, 24(2):121–129, 1998.

C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID, 2004*, 2004.

D. Ponceleon, S. Srinivasan, A. Amir, D. Petkovic, and D. Diklic. Key to effective video retrieval: Effective cataloguing and browsing. In *Proc. ACM Multimedia*, pages 99–107, 1998.

K. Porkaew, K. Chakrabarti, and S. Mehrotra. Query refinement for multimedia similarity retrieval in MARS. In *Proc. of the 7th ACM Int. Conf. on Multimedia*, pages 235–238, Orlando, Florida, USA, 1999.

Redaktion. Die Abschaffung der Filmrolle. *Der Spiegel*, 35:70 ff., 2003.

S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *Journal of Documentation*, 46(4):359–364, 1990.

J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, Englewood Cliffs, USA, 1971. Prentice-Hall.

F. Roeder. Carl Friedrich Gauss. *Backsights*, 1993.

R. Russell, D. Quinlan, and C. Yeoh. *Filesystem Hierarchy Standard*. Filesystem Hierarchy Standard Group, 01 2004.

I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 213–220, New York, NY, USA, 2003. ACM Press.

D. F. Salisbury. Memorial Service May 20 for Arthur Schawlow, laser co-inventor. *Stanford Report*, 5 1999.

G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. *ACM SIGIR conference on research and development in Information Retrieval*, pages 49–58, 1993.

P. Schaff and H. Wace, editors. *St. Ambrose: Select Works and Letters (Nicene and Post-Nicene Fathers*. Wm. B. Eerdmans Publishing Company, 09 2002.

J. Shannon. YouTube's Interface: If You Build It, They Will Come. *Online Media Daily*, 4 2006.

L. Slaughter, G. Marchionini, and G. Geisler. Open Video: A Framework for a Test Collection. In *Journal of Network and Computer Applications, Special Issue On Network-Based Storage Services*, pages 219–245, 2000.

A. F. Smeaton. The Físchlár Digital Library: Networked Access to a Video Archive of TV News. In *TERENA Networking Conference 2002, Limerick, Ireland, 3-6 June 2002*, 2002.

A. F. Smeaton, J. Gilvarry, G. Gormley, B. Tobin, S. Marlow, and N. Murphy. An evaluation of alternative techniques for automatic detection of shot boundaries in digital video. In *IMVIP 1999 – Irish Machine Vision and Image Processing Conference. Dublin, Ireland*, 1999.

A. F. Smeaton and T. Ianeva. TRECVID-2005: Search Task. In *TREC video retrieval evaluation online proceedings*, 2005.

A. F. Smeaton, P. Over, and J. Arlandis. TRECVID-2004: Search Task Overview. In *TREC video retrieval evaluation online proceedings*, 2004.

A. F. Smeaton, P. Over, and R. Taban. The TREC-2001 Video Track Report. In E. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text Retrieval Conference TREC-10*, pages 52–60, Gaithersburg, Maryland, USA, 2002. Department of Commerce, National Institute of Standards and Technology.

A. F. Smeaton and P. Wilkins. TRECVID 2004: Interactive Search Questionnaires. `http://www-nlpir.nist.gov/projects/tv2004/questionnaires.html`, 09 2004. last checked: 26.07.2006.

C. G. M. Snoek, M. Worring, J. M. Geusebroek, D. C. Koelma, and F. J. Seinstra. The MediaMill TRECVID 2004 Semantic Video Search Engine. In *TRECVid retrieval evaluation online proceedings*, 2004.

F. Souvannavong, B. Merialdo, and B. Huet. Eurécom at TRECVid 2004: Feature Extraction Task. In *TREC2004 – Text REtrieval Conference, Gaithersburg, Maryland, 15-19 November 2004*, 2004.

U. Srinivasan, L. Gu, K. Tsui, and W. G. Simpson-Young. A data model to support content-based search in digital videos. *The Australian Computer Journal*, 29(4), 11 1997.

Sun Microsystems. Simple API for XML. `http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPSAX.html`, 2006. last checked: 25.05.2006.

S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 107–118. ACM Press, 2001.

J. Urban and J. M. Jose. EGO: A personalised multimedia management tool. In *Proc. of the Second International Workshop on Adaptive Multimedia Retrieval*, 2004.

C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

Voltaire. *Candide*. Bantam Classics, New York, USA, 01 1984.

J. Watkinson. *The MPEG Handbook*. Focal Press, 09 2001.

J. Weber. YouTube: the good, the bad and the interesting. *The Times Online*, 6 2006.

R. W. White, J. M. Jose, and I. Ruthven. The use of implicit evidence for relevance feedback in Web retrieval. In *Proceedings of 24th ECIR Conference*, pages 93–109, 2002.

B. M. Wildemuth, G. Marchionini, M. Yang, G. Geisler, T. Wilkens, A. Hughes, and R. Gruss. How fast is too fast?: evaluating fast forward surrogates for digital video. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 221–230, Washington, DC, USA, 2003a. IEEE Computer Society.

B. M. Wildemuth, M. Yang, A. Hughes, R. Gruss, G. Geisler, and G. Marchionini. Access via Features versus Access via Transcripts: User Performance and Satisfaction. In *TREC2003 – Text REtrieval Conference, Gaithersburg, Maryland*, 11 2003b.

S. Woolley. Raw and Random. *Forbes*, page 27, 3 2006.

J. Xu. *Solving the word mismatch problem through automatic text analysis*. PhD thesis, University of Massachusetts at Amherst, 1997.

J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

R. Yan, J. Yang, and A. G. Hauptmann. Learning Query-Class Dependent Weights in Automatic Video Retrieval. In *Proceedings of ACM Multimedia 2004, New York, USA*, pages 548–555, 10 2004.

M. Yang and G. Marchionini. Exploring Users' Video Relevance Criteria – A Pilot Study. In *ASIST 2004 Annual Meeting; Managing and Enhancing Information: Cultures and Conflicts*, Providence, Rhode Island, USA, 11 2004.

M. Yang and G. Marchionini. Deciphering visual gist and its implications for video retrieval and interface design. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1877–1880, New York, NY, USA, 2005. ACM Press.

M. Yang, B. M. Wildemuth, and G. Marchionini. The relative effectiveness of concept-based versus content-based video retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 368–371, New York, NY, USA, 2004. ACM Press.

M. Yang, B. M. Wildemuth, G. Marchionini, T. Wilkens, G. Geisler, A. Hughes, R. Gruss, and C. Webster. Measuring User Performance During Interactions with Digital Video Collections. In *ASIST 2003 Annual Meeting – Humanizing Information Technology: From Ideas to Bits and Back*, Westin Long Beach, California, USA, 10 2003.

B. Yeo and M. Yeung. Retrieving and visualizing video. *Communications of the ACM*, 40(12): 43–52, 1997.

YouTube. Categories. `http://www.youtube.com/categories`, 2006a. last checked: 23.06.2006.

YouTube. Help Center. `http://www.youtube.com/t/help_center`, 2006b. last checked: 23.06.2006.

Y. Zhai, J. Liu, and M. Shah. Automatic Query Expansion In News Video Retrieval. *International Conference of Multimedia and Expo, Toronto, Canada*, 2006.

H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

R. Zhao and W. Grosky. Bridging the Semantic Gap in Image Retrieval. In T.K. Shih, editor, *Distributed Multimedia Databases: Techniques and Applications*, pages 13–36, Hershey, Pennsylvania, USA, 2002. Idea Group Publishing.