Semantics of Video Shots for Content-based Retrieval

A thesis submitted for the degree of
Doctor of Philosophy

Dipl.-Ing. Timo Volkmer,
School of Computer Science and Information Technology,
Science, Engineering, and Technology Portfolio,
RMIT University,
Melbourne, Victoria, Australia.

May 4, 2007
Declaration

I certify that, except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Timo Volkmer
School of Computer Science and Information Technology
RMIT University
May 4, 2007
Acknowledgments

“I may not have gone where I intended to go, but I think I have ended up where I needed to be.”

– The Salmon of Doubt by Douglas Noël Adams

I thank my family and friends in Germany who have constantly been supporting and encouraging me, despite the fact that I have deserted them to spend over four years on another side of our planet, pursuing my own goals.

I am very grateful to my first supervisor Dr. Seyed M. M. “Saied” Tahaghoghi for introducing me to the world of academic research and teaching. Saied has been an inspiration, professionally and as a friend. Many thanks appertain my second supervisor, Associate Professor James A. Thom, whose professional criticism and advice I have always appreciated, even when it meant another long night in front of the computer for me. I believe it is this kind of positive criticism and support that makes us deliver high quality research.

I offer my deep gratitude to the members of the Intelligent Information Analysis Group at the IBM T. J. Watson Research Center in Hawthorne, New York, for providing me the opportunity of working with them. I thank my ever-diligent mentor Apostol “Paul” Natsev for being a role-model in conducting professional research, and for his advice on Bulgarian feta cheese. I also thank my managers at IBM, Murray S. Campbell and John R. Smith, and my fellow IBM team members Jelena Tesić and Milind R. Naphade for being such inspiring colleagues and making my time in New York both professionally and personally a marvellous experience.

I am very thankful to my fellow students of the Search Engine Group at RMIT University (in no particular order) Vaughan Shanks, Falk Scholer, Jovan Pehcevski, Ranjan Sinha, Steven Garcia, Halil Ali, and all the others who have provided this enjoyable work environment. It has truly been a unique and incredibly rewarding experience to be a part of this group, and I will never forget our enlighting discussions on (research-)life, the Universe, and everything.

My sincere thanks also go to all my friends from the RMIT Outdoors Club and from the RMIT Underwater Club for incessantly reminding me that there is more to life than just computer science research, and that the beautiful state of Victoria in Australia is indeed The Place to Be.

Last but not least, I thank all anonymous reviewers of previous publications who, by giving their valued comments, contributed to the quality of my work.
Credits

Portions of the material in this thesis have previously appeared in the following publications:

Journal Article


Refereed Conferences and Workshops


Unrefereed Workshops


This work was supported by the School of Computer Science and Information Technology at RMIT University and the Australian Research Council (ARC)\(^1\). Parts of this work have been conducted while the author was visiting the IBM T. J. Watson Research Center in Hawthorne, NY, USA. The thesis was typeset using the \LaTeX\ document preparation system. All trademarks are the property of their respective owners.

Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

\(^1\)http://www.arc.gov.au
Contents

Abstract 1

1 Introduction 3
  1.1 Research Objectives and Methodology 5
  1.2 Thesis Overview 8

2 Fundamental Aspects of Content-based Video Retrieval 10
  2.1 Structure and Representation of Digital Video 11
      2.1.1 Frames, Shots, and Transitions 11
      2.1.2 On the Representation of Colour 13
      2.1.3 Feature Representation 17
      2.1.4 Distance Metrics 24
  2.2 Temporal Segmentation 27
      2.2.1 Review of Existing Techniques 28
      2.2.2 A Ranking Approach to Cut Detection 32
      2.2.3 Systematic Performance Evaluation 34
  2.3 Semantic Classification 36
      2.3.1 Automatic Classification 37
      2.3.2 Manual Annotation 40
  2.4 Content-based Video Search 42
      2.4.1 Video Search using Spoken Text 42
      2.4.2 Query Expansion for Speech-based Video Search 44
      2.4.3 Multimodal Search 47
      2.4.4 Performance Evaluation 48
  2.5 Summary 50
3 Temporal Video Shot Segmentation ............................................. 52
  3.1 Improvements on Histogram-based Cut Detection ................. 53
    3.1.1 Low-level Feature and Distance Metric Selection ............ 53
    3.1.2 Dynamic Threshold Computation ...................................... 55
  3.2 Gradual Transition Detection using Average Frame Similarity .... 56
  3.3 Algorithm Parameters .......................................................... 60
  3.4 Performance Evaluation .......................................................... 61
    3.4.1 Methodology ..................................................................... 61
    3.4.2 Experiments and Discussion .............................................. 62
    3.4.3 Comparison with other Approaches ..................................... 70
  3.5 Summary .................................................................................. 73

4 Modelling Human Judgement of Digital Imagery ......................... 74
  4.1 Large-scale Annotation of Visual Documents ....................... 75
  4.2 Modelling Multiple Judgements ................................................. 77
    4.2.1 Latent Class Modelling ..................................................... 81
    4.2.2 Estimating Annotation Quality ....................................... 87
    4.2.3 Assessing Goodness-of-fit .............................................. 90
  4.3 Discussion of Results ............................................................... 90
  4.4 Summary .................................................................................. 97

5 Manual Annotation of Visual Content ........................................ 99
  5.1 The TRECVID 2005 Annotation Effort at IBM ....................... 100
    5.1.1 The IBM EVA Annotation System ................................... 103
    5.1.2 Data Evaluation and Results ..................................... 107
    5.1.3 Considerations on the TRECVID Results ....................... 113
  5.2 Annotation Experiments at RMIT University ....................... 115
    5.2.1 Objectives and Methodology ..................................... 116
    5.2.2 Evaluation and Results ............................................... 118
  5.3 Summary .................................................................................. 127

6 Content-based Video Search ..................................................... 130
  6.1 Using Spoken Text for Video Search .................................... 131
    6.1.1 Query Refinement .......................................................... 132
    6.1.2 Effectiveness of Query Refinement Techniques ............... 132
6.1.3 Experiments and Evaluation ........................................ 135
6.2 A Fusion Approach to Content-based Video Search ................. 141
  6.2.1 The Sapphire Search System ..................................... 142
  6.2.2 Adding Semantic Concept Terms ................................. 143
  6.2.3 Semantic Concept Term Expansion ............................... 145
  6.2.4 Query Preparation and Analysis .................................. 146
  6.2.5 Training Experiments with the TRECVID 2005 Test Set ........ 147
  6.2.6 TRECVID 2006 Experiments ...................................... 149
6.3 Summary ...................................................................... 155

7 Conclusions and Future Work ............................................. 157
  7.1 Lessons Learned .......................................................... 158
  7.2 Applications and Future Work ....................................... 160


B TRECVID 2005 Annotation Forum Concepts ............................ 167

C Selected TRECVID Search Corpora ...................................... 170

Bibliography ...................................................................... 173
## List of Figures

2.1 General structure of a video ........................................... 11
2.2 The HSV colour cone ................................................... 16
2.3 An example video frame and its 3D colour histogram in the HSV colour space 18
2.4 Dividing a video frame into regions to extract a localised histogram ............. 20
2.5 An example video frame before and after applying an edge detection filter . . 22
2.6 Selected Minkowski norm distances in comparison ............................ 25
2.7 An example of separating positive and negative examples in two dimensions . 37
2.8 Schema of a state-of-the-art multimodal video search engine ................... 47
3.1 Dividing a video frame into regions to extract a localised histogram .......... 54
3.2 A moving window of consecutive frames traversing a dissolve transition ....... 57
3.3 A schematised dissolve transition ..................................... 58
3.4 Plot of $R_{pp}$ over a 200-frame interval of video ............................ 60
3.5 Varied half-window sizes $W_g$ for gradual transition detection ............... 66
3.6 Varied half-window sizes $W_c$ for cut detection ................................ 67
3.7 Varied threshold factors $T$ for gradual transition detection ..................... 68
4.1 Positive ratings for the image/concept pairs with one judgement ............... 78
4.2 Positive ratings for the image/concept pairs with two judgements ............... 79
4.3 Positive ratings for the image/concept pairs with three judgements ............. 80
4.4 Positive ratings for the image/concept pairs with four judgements ............. 81
4.5 Positive ratings for the image/concept pairs with five judgements ............. 82
4.6 All ratings that we obtained for concept Bus .................................. 83
4.7 The number of positive examples compared to the model’s prediction ........ 86
4.8 Classification index $\lambda$ compared to intraclass correlation $\kappa$ ................. 89
4.9 Example classifications for concept Bus ..................................... 92
List of Tables

3.1 Shot boundary detection results for the TRECVID test collections . . . . . . . 63
3.2 Analysis of false positive and false negative detections . . . . . . . . . . . 64
3.3 Detailed analysis of gradual transition detection performance . . . . . . . 65
3.4 Comparative timing results for the TRECVID 2004 test set . . . . . . . . . 69
3.5 Comparative results of the TRECVID shot boundary detection task . . . . . 71

4.1 Goodness of fit evaluation for concept Studio . . . . . . . . . . . . . . . . . . . 91
4.2 Final result of LCM-based classification . . . . . . . . . . . . . . . . . . . . . . 94

5.1 The semantic concepts used in the TRECVID 2005 annotation effort . . . . . 102
5.2 RMIT annotation experiment concept sets a and b . . . . . . . . . . . . . . . 116
5.3 LCM-based concept prevalences for the annotation experiment at RMIT . . . 120

6.1 TRECVID 2005 search test set query definitions . . . . . . . . . . . . . . . . . . . 135
6.2 Mean average precision scores of the text search baseline . . . . . . . . . . . . 138
6.3 Query-class specific mean average precision scores for the baseline . . . . . . 140
6.4 TRECVID 2005 search test set query definitions . . . . . . . . . . . . . . . . . . . 151
6.5 Average Precision results for the TRECVID 2006 collection . . . . . . . . . . 153
6.6 Precision at 20 returned results on the TRECVID 2006 collection . . . . . . . 154
Content-based video retrieval research combines expertise from many different areas, such as signal processing, machine learning, pattern recognition, and computer vision. As video extends into both the spatial and the temporal domain, we require techniques for the temporal decomposition of footage so that specific content can be accessed. This content may then be semantically classified — ideally in an automated process — to enable filtering, browsing, and searching. An aspect of visual information retrieval that has been largely neglected is the consideration of varying interpretations of visual content by users. Visual information is usually perceived and interpreted differently by individual users because pictorial representation of information is often less specific than its textual representation.

In this thesis, we address several fundamental issues of content-based video retrieval for effective handling of digital footage. Temporal segmentation, the common first step in handling digital video, is the decomposition of video streams into smaller, semantically coherent entities. This segmentation is usually performed on the basis of detecting the transitions that are edited into the video to combine single camera takes. While abrupt transitions — cuts — can be detected relatively well with existing techniques, effective detection of gradual transitions remains difficult. We present our approach to temporal video segmentation, proposing a novel algorithm that evaluates sets of frames using a relatively simple histogram feature. Our technique has been shown to range among the best existing shot segmentation algorithms in large-scale evaluations.

The next step is the semantic classification of each video segment to generate an index for content-based retrieval in video databases. Machine learning techniques can be applied effectively to classify video content. However, these techniques require manually classified examples for training before automatic classification of unseen content can be carried out. The process of manually classifying training examples is not trivial because of the implied ambiguity of visual content. We propose an unsupervised learning approach based on latent
class modelling in which we obtain multiple judgements per video shot and model the users’ response behaviour over a large collection of shots. This technique yields a more generic classification of the visual content. Moreover, it enables the quality of the classification to be assessed, and maximises the number of training examples by resolving disagreement between judgements. We apply this approach to data obtained in a large-scale, collaborative annotation effort and present ways to improve the effectiveness for manual annotation of visual content by better design and specification of the process.

The availability of automatic speech recognition helps to implement text-based video search. Together with semantic classification of video content, this technique can be used to implement video search using free-text queries. This requires the application of text search techniques to video and the successful combination of information sources. We explore several text-based query expansion techniques for speech-based video retrieval, and propose a fusion method to improve the effectiveness of text-based video retrieval. To combine both text and visual search approaches, we explore a fusion technique that combines spoken information and visual information using semantic keywords automatically assigned to the footage based on the visual content.

The techniques that we propose in this thesis help to facilitate effective content-based video retrieval and highlight the importance of considering different user interpretations of visual content. This allows better understanding of video content and a more holistic approach to multimedia retrieval in the future.
Chapter 1

Introduction

“You insist that there is something that a machine can’t do. If you tell me precisely what it is that a machine cannot do, I can always make a machine which will do just that.”

– John von Neumann

Humans perceive their environment largely by audiovisual means. The human brain and visual system together with the human auricular skills provide outstanding capabilities to process audiovisual information, and instantly interpret its meaning on the basis of experience and prior knowledge. Audiovisual sensation is the most convenient and most effective form for humans to consume information — we believe what we can see and hear, and we prefer to share our experiences by aural and visual description. In particular for complex circumstances, visualisation is known to convey the facts of the matter best.

Sharing and recording experiences and knowledge has always been an important aspect of human evolution, contributing to the enhancement of knowledge of current and future generations. Our means of communication have constantly become more sophisticated. Television and the Internet are the latest inventions that have enabled the most efficient sharing of information that human kind has ever seen. Television has brought access to produced video content to billions of consumers; on the Internet, consumers become authors and distribute written documents and pictures as well as sound and video files around the globe. Among others, Microsoft founder and Chairman Bill Gates envisions the fusion of Internet and Television to become the next generation video platform [Gates and Bach, 2007].

1Indeed, Bill Gates’ keynote at the 2007 Consumer Electronics Show (CES) can be viewed as an online video: http://www.microsoft.com/winme/0701/29031/CES.asx.
Video as a carrier of information plays an important role in sharing information today. The most important benefit of video is its capacity to convey information in a way that serves the human desire best to perceive and consume information by audiovisual means. Naturally, the amount of footage that is produced, distributed, and stored today is enormous.

As of 2003, the archives of one of Europe’s largest television broadcasters, Radio Tele Luxembourg (RTL), were growing by 700 GB every year [Sietmann, 2003]. According to Sietmann [2003], the archive of the British media organisation BBC held approximately 750,000 hours of video in 2003. The BBC has been employing 30 full-time staff to catalogue new content and handle the up to 3,000 content requests per week.

As soon as network bandwidth, computing power, and storage capabilities allowed transmitting images on the Internet, we witnessed a world-wide spread of community applications to share photos. Following this trend, we now see growing use of Internet applications such as YouTube\(^2\) and Yahoo! Video\(^3\) that allow sharing video across the world. While being very efficient in distributing information world-wide, the Internet lacks mechanisms for effective data organisation and management. Retrieving specific video content that meets a given information need is seriously hampered because the content is not adequately indexed. Many video sharing applications on the Internet feature *community tagging*, which enables users to label video clips with more or less descriptive textual information. These textual labels can be used to retrieve footage using search engines, but the benefit of these labels is questionable as they are often incomplete and highly subjective. Moreover, such tags can currently only be assigned at the file level and users cannot access a specific segment without downloading and reviewing the entire video. Ideally, we should be able to retrieve a specific part of a video that meets our information need by specifying what this part of the video shows, similar to referencing a particular paragraph on a specific page in a book. This desire is even stronger when storing video in large proprietary archives with the need for frequent retrieval. For example, this is the case for footage from surveillance, video conferencing, and broadcast television. For such databases to be useful, we require effective mechanisms to analyse and filter large amounts of video content.

Clearly, without appropriate organisation and management, video content in digital libraries is of only limited utility. For accessing specific sections of a video that meet an information need, we must be able to decompose video streams into smaller entities and describe their semantics so that we can build an index for effective retrieval. Methods for

\(^2\)http://www.youtube.com  
\(^3\)http://video.yahoo.com
CHAPTER 1. INTRODUCTION

automatic video segmentation and classification should be generic and scalable so that they can be applied to large and diverse data collections.

While research in recent years has focused on providing technical solutions to automatic video classification, the human factor has often been neglected. In particular, the fact that visual information leaves much room for interpretation and ambiguity causes problems when designing automatic classification algorithms. We frequently observe disagreement when humans describe visual content; this should raise the question how good a technical solution to this problem can be if not even humans can agree unambiguously on a description of visual content. This consideration should have an impact on current and future retrieval techniques for visual information.

1.1 Research Objectives and Methodology

The research presented in this thesis addresses several fundamental issues when accessing, classifying, and retrieving digital footage. Our work has included regular participation in the annual TREC Video Retrieval Evaluation (TRECVID)\textsuperscript{4} \cite{Smeaton2005}, a research workshop organised and funded by the U.S. American National Institute of Standards and Technology (NIST),\textsuperscript{5} facilitating research in video retrieval. Not surprisingly, the BBC is one of the main supporters of TRECVID \cite{Kraaij2006,Over2005}.

This participation has allowed us to model realistic problem settings by evaluating our techniques using large test corpora that primarily consist of television news broadcasts. By using the common TRECVID test-bed and evaluation metrics, we make our results comparable to those of other participants. The TRECVID evaluation uses a common reference, also called ground-truth, that specifies which items in a collection are considered relevant to a specific information need. Given this ground-truth, the performance of approaches is measured with the common information retrieval metrics recall and precision \cite{vanRijsbergen1979}. Recall is a measure of how complete a result is, that is how many relevant items have been found in comparison to the total amount of relevant items. Precision measures how correct a result is by considering the amount of items that were correctly found in comparison to the total amount of items that were found. Ideally, both recall and precision are at 100% but this can in practice usually not be achieved as recall and precision tend to exhibit opposing behaviour. For example, a system may be optimised to achieve high recall by returning

\textsuperscript{4}http://www-nlpir.nist.gov/projects/trecvid
\textsuperscript{5}http://www.nist.gov.au
very many possibly correct results but this also increases the likelihood of returning many incorrect results which hurts precision. These and other derived evaluation metrics that we use are explained in detail in the next chapter.

A first step in providing effective access to video content is to store footage in a way that allows us to retrieve the parts that concern our information need. An obvious way to achieve this is to separate video streams into smaller sequences, that can be stored and indexed individually. In the video production process, individual shots are concatenated in an editing process, where a shot refers to a single camera take [Konigsberg, 1989]. Some form of transition effect is usually applied to concatenate the shots in a video. We distinguish abrupt transitions — also called cuts — that represent an immediate shot change between two camera takes, and gradual transitions in which one shot gradually changes into the next shot over some time. Because transitions form the boundaries between adjacent shots, we refer to these as shot boundaries. Dictated by this production process, the common approach is to divide videos into their constituent shots in preparation for storage, classification, and retrieval.

The process of shot segmentation is usually performed by identifying the transition effects that have been edited into the video during production. This process is not trivial given that modern footage may contain a variety of complex transition effects. In particular, the detection of gradual transitions is a difficult problem, which we address in the following research question:

**Can we develop an effective algorithm for gradual transition detection?**

After a video has been temporally segmented into shots, we aim to build an index that describes the semantics of each shot so that we can effectively search and retrieve content using free-text queries. Manual annotation, the process of assigning descriptive textual labels to video content, is a possible approach but it is often inconsistent and not scalable to large databases. The inconsistency of manual content description shows, however, that different human consumers perceive audiovisual content differently because it leaves room for individual interpretation. Shatford [1986] concludes that the interpretation of pictures cannot be consistently indexed. Recent research in content-based video retrieval applies machine learning techniques for automatic semantic classification of video shots. This approach shows promising results but requires the machine learning algorithms to be trained with example video shots that have previously been manually classified. This means that the initial problem of indexing ambiguous content remains because we need to find a way to manually
classify audiovisual content even when users disagree on it. The advancements that machine learning techniques have provided to automatic video classification motivate investigating the problem of classifying ambiguous visual content, which we address in the following research question:

**How can we combine multiple, non-uniform user opinions on visual documents to generate accurate training data for video retrieval?**

The process of generating such training data has in recent years been approached by organising collaborative efforts which aim to manually annotate a given video corpus with descriptive textual labels. It has been shown to be useful to define a limited vocabulary of textual labels — semantic concepts — to promote the consistency and the completeness of the annotations. Initial results of such annotation efforts suggest that the quality of the annotations varies substantially. In particular, when we collect multiple judgements per video shot, we frequently observe varying levels of disagreement for different semantic concepts. This suggests that some concepts are more difficult to annotate than others. Moreover, since manual annotation is a time-consuming and error-prone task, we need to carefully specify the effort so that manual labour is effectively utilised. While computer programs are employed to support the annotation task, very little is known about how these should best be designed. We investigate the problems of what to annotate and how to annotate it, combined in the following research question:

**How can we specify the annotation mode and concept vocabulary to maximise annotation quality while maintaining efficiency?**

The techniques that we have outlined thus far lead to semantic indexes that use text to describe video content with the aim of enabling concept-based video search. This is complemented by successful research on automatic speech recognition techniques that can be used to reliably transcribe spoken information from video into text. By incorporating spoken text into the content-based search approach, we can increase its effectiveness. However, we frequently observe that text-based video search is less effective than text search on written documents. We address this issue and explore techniques that analyse and reformulate user queries — known as query expansion — to improve the effectiveness of video search. As a second aspect, we investigate the application of such a technique at indexing time. We generate a combined index containing the spoken text and semantic concept terms describing the visual content. Instead of reformulating the query, we expand the concept terms in
the index with semantically related terms. We therefore investigate the following research question:

**Can term expansion techniques be used to improve content-based video retrieval?**

These research questions concern several vital aspects of handling digital footage, spanning tasks from low-level processing to semantic classification and search. Thus the work in this thesis contributes to several aspects of research progress in the field of content-based video retrieval.

### 1.2 Thesis Overview

The remainder of this thesis is organised as follows. In Chapter 2, we introduce the general representation of digital video and how we can access its content for analysis. We discuss the current state of the art in video shot segmentation, automatic and manual classification, and content-based search as a motivation for the research questions addressed in the subsequent chapters.

In Chapter 3, we present our approach to video shot segmentation that includes improvements upon an existing method for cut detection. The key contribution, however, is our novel method for effective gradual transition detection. We combine both methods in one algorithm and discuss experimental results on several large collections. To demonstrate the effectiveness of our approach, we compare our results with those of other TRECVID participants.

We address the problem of classifying visual content by modelling multiple user judgements of video shots in Chapter 4. To model a more generic view of the semantics contained in visual information, we propose an unsupervised learning approach using latent class modelling. This enables unambiguous classification of video shots based on multiple, non-uniform judgements, and helps to generate high-quality training data for machine learning algorithms.

In Chapter 5, we present the evaluation and outcomes of two user studies on collaborative annotation efforts. We formulate guidelines to improve the process of manual image and video annotation. Our guidelines help to optimise the resource utilisation and the effectiveness when manually annotating large corpora, forming an important contribution to generating high-quality training data for video retrieval purposes.
We explore several term expansion techniques for content-based video search in Chapter 6. The contribution of this chapter is two-fold. First, we propose a combination approach for applying several term expansion techniques to the user queries. We show that this technique improves the overall precision of speech-based video search across a range of different query topics. Second, we discuss a term expansion approach applied at indexing time to combine spoken text and concept terms from visual content for video search. While this method does not consistently achieve significant improvements across all our test experiments, our results highlight the importance of intelligent information fusion and query processing techniques to video retrieval.

In Chapter 7, we summarise our findings and present our overall conclusions, including an outlook on applications and future work.
Chapter 2

Fundamental Aspects of
Content-based Video Retrieval

“If we knew what it was we were doing, it would not be called research, would it?”
– Albert Einstein

Video retrieval is a relatively young, but active field of research in which many of the techniques that are used have been derived and extended from still image processing. As a result, numerous approaches for representing and accessing digital video content exist.

In this chapter, we review fundamental techniques for handling digital footage. We discuss the current state-of-the-art in content-based video retrieval that builds the foundations for our research. Focusing on approaches concerning temporal video segmentation, video classification, indexing, and search, we identify shortcomings to motivate for our research that we present in the remainder of this thesis.

We describe the general structure of digital video and how it is digitally represented in Section 2.1. In Section 2.2, we discuss temporal video segmentation as one of the primary processing steps for handling digital footage. We then describe the main aspects of automatic semantic video classification in Section 2.3 as a motivation to investigate issues related to manual video annotation and content modelling more closely. Approaches to content-based video search that combine many of the fundamental techniques are discussed in Section 2.4.
Figure 2.1: A video is composed of individual images — the frames — that are replayed at a certain speed to create the effect of continuous motion. Several frames form shots, which are in turn combined to form stories, episodes, or scenes that the video is composed of.

2.1 Structure and Representation of Digital Video

Figure 2.1 shows an illustration of the general structure of a video. In this thesis, we focus on digital video, and are not concerned with analogue representations.

2.1.1 Frames, Shots, and Transitions

We define the frame as the smallest semantic entity in a video. Frames are images that represent the visual content of the video at a specific time, containing only spatial information and no temporal information. We can describe an image as a picture that is digitally represented through discrete pixels each with a specific location and colour [Hampapur et al., 1994]. For example, let $I[X,Y]$ specify an $X \times Y$ sized image, so that $I_{xy} = c_k$ is the pixel at location $(x, y)$ with the colour $c_k$, where $x \in \{0, \ldots, X - 1\}$ and $y \in \{0, \ldots, Y - 1\}$. The colour $c_k$ is a specific colour out of the codebook $C = \{c_0, \ldots, c_k, \ldots, c_{K-1}\}$ of $K$ available colours. We describe the representation of colour in more detail below.

Several frames recorded in succession form a shot, also referred to as a camera take. A shot is a sequence of frames that presents visual content recorded in a single, uninterrupted camera operation [Konigsberg, 1989]. Individual shots are then concatenated using transition effects.
in the editing process to form a video clip. The literature [Del Bimbo, 2001; Konigsberg, 1989] also defines stories, episodes, and scenes, which consist of one or more shots. However, the use and definition of these terms is often inconsistent and most of these terms are not of relevance to our work. For our purposes, besides the shot, we merely define a story as one or more successive shots with a common focus on a specific topic.

The speed at which frames are recorded (and ideally replayed) is called as the frame rate. For example, the European PAL\(^1\) standard defines a frame rate of 25 frames per second. The U.S. American NTSC\(^2\) standard defines a frame rate of 29.97 frames per second.

Video files generally contain an audio track that contains the sound that was recorded together with the visual information. The sound is stored in digitised form and synchronised with the visual content [Konigsberg, 1989]. Both sound and visual tracks are usually compressed for storage and transmission as the amount of data to handle would otherwise be too large for practical purposes. The best-known and most widely used video compression standards are those of the MPEG family. The Moving Pictures Expert Group (MPEG)\(^3\) is an expert consortium that specifies standardised formats for the digital representation of video. For example, the MPEG-1 standard [ISO/IEC, 1996] is used to encode footage for the Video CD (VCD) format, and is popular for Internet video. The more advanced MPEG-2 standard [ISO/IEC, 2000] is common in professional broadcast technology and is used for encoding DVD-Video. MPEG-2 can be seen as an extension of MPEG-1, specifying more advanced, optional features. The main features of both MPEG-1 and MPEG-2 are spatial compression that is applied to each individual frame as well as compression in the temporal domain to reduce redundancy between successive frames [Chen, 2000; Watkinson, 2001].

The spatial compression scheme used in these MPEG standards is the same that is used in the JPEG [ISO/IEC, 1992] image compression standard. This defines Macro Blocks (MB) that consist of $8 \times 8$ pixels and applies the Discrete Cosine Transform (DCT) [Ahmed et al., 1974] to each block. Data reduction is achieved by quantising the DCT coefficients and by removing all those that are zero or near-zero. This reduction of spatial information within each frame is also referred to as \textit{intra}(-frame) coding [Chen, 2000]. The reduction of temporal redundancy is termed \textit{inter}(-frame) coding [Chen, 2000] and exploits the fact that adjacent

\(^1\)Phase Alternating Line (PAL) is a colour encoding system used in broadcast television systems mostly in Europe, Africa, Asia, and Australia.

\(^2\)An analog broadcast television standard by the National Television Systems Committee (NTSC) that is used in the U.S.A. and other countries in the Americas.

\(^3\)http://www.chiariglione.org/mpeg
frames in a video are often similar with only small changes in a depicted scene. The principle of inter coding is to start encoding using the full (only spatially compressed) information for the first frame and include only the difference to this frame when encoding subsequent frames. The content for the following frames is then approximated based on the assumption that the macro blocks from the first frames have only slightly moved. Each macro block is fitted in the following frames and displacement between the previous frame and the following frame is computed. A frame can thus be reconstructed using the previous frame and the displacement for each macro block. The displacement for a macro block is also called motion vector [Watkinson, 2001]. A frame that is compressed using intra coding is called an I-frame; frames that are encoded with inter coding are called predicted frames, or P-frames.

The terms R-frame and key frame are frequently used in video retrieval [Flickner et al., 1995; Heesch et al., 2003; Pickering et al., 2002]. These terms do not refer to the encoding but rather describe a frame that is used to represent the visual content of an entire video segment. For example, we may select one frame to approximately represent the visual content of a shot [Flickner et al., 1995]. This process is called key frame extraction and can be performed by identifying the frame at the centre of a shot, and then selecting the nearest I-frame. This I-frame is then used as the key frame because I-frames are usually of higher quality than images that are reconstructed from a P-frame.

2.1.2 On the Representation of Colour

For digitally representing colour, we must define a colour model so that we can mathematically describe colours. While the terms colour model and colour space are often used interchangeably, there is a significant difference between them: a colour model describes how colours can mathematically be represented but does not consider how they can be reproduced with different devices under different conditions [Hunt, 2004]. Only in combination with a definition of the viewing conditions, a colour model can be used to define a colour space. A widely known colour model is the RGB model in which each colour is represented by a blue, green, and red component. This is based on the ability of the human visual system to distinguish between short, middle, and long wavelengths of light, representing the blue, green, and red components of white light [Hunt, 2004].

The Commission Internationale d’Eclairage (CIE)\(^4\) defined the CIE XYZ colour space, also known as the CIE 1931 colour space, to serve as an absolute colour space. Its definition

\(^4\)English translation: International Commission on Illumination — http://www.cie.co.at/cie
relates back to experiments involving several human observers [Guild, 1931; Wright, 1928].

The CIE XYZ colour space is an absolute colour space because it is device independent and contains all colours that an average human can perceive. This is also referred to as the human colour gamut. The colour gamut — or simply the gamut — is the complete sub-set of colours that can be represented with a given method. The experiments by Wright [1928] and Guild [1931] also resulted in the definition of the CIE standard observer and a range of standard viewing conditions, known as standard illuminants, or white points [Hunt, 2004]. For example, the D65 standard illuminant simulates a sunny day in the northern hemisphere.

The CIE XYZ colour space, the CIE standard observer, and the standard illuminants are the basis for nearly all mathematical representations of colour as of today.

The CIE 1976 (L*,a*,b*) colour space, also called CIELAB or simply LAB colour space, is based on the CIE XYZ colour space and serves as a mathematical model for perceptually uniform colour difference operations. Perceptual uniformity refers to the effect of perceived differences between colours being proportional to the geometric differences of colours in the given colour space [Hunt, 2004]. The CIELAB colour space is device independent and its gamut is even larger than the human gamut, including colours that cannot be physically reproduced by any device. This colour space is often used in image editing applications to enable perceptually uniform operations. However, this is not of importance for our work and therefore we do not consider this colour space here.

When reproducing colour using a device such as a printer, a camera, or a computer monitor, we must choose a colour model and consider the colour reproduction capabilities of the device. For example, using the RGB colour model at standard illumination conditions D65 to reproduce colour on a CRT television screen, we can define an RGB colour space on the basis of the CIE XYZ colour space as follows [Hunt, 2004]:

\[
R = 0.607X + 0.299Y + 0.000Z \\
G = 0.174X + 0.587Y + 0.066Z \\
B = 0.200X + 0.114Y + 1.111Z
\]  \hspace{1cm} (2.1)

The coefficients in these equations reflect the nonlinear ability of a CRT screen to reproduce colour under the D65 viewing conditions. A standardised colour space for the reproduction of colour using modern LCD screens with the RGB colour model is the sRGB [ISO/IEC, 1999] colour space. The definition of this colour space considers average viewing conditions that are appropriate for most use cases.
The standard colour space used for analogue video signals such as broadcast television is the YUV colour space. In this colour space, the Y component represents the luminance and the U and V components represent the chrominance in terms of the difference to the blue colour component (U) and the red colour component (V). The YUV colour space can be seen as a way of encoding RGB colour for television broadcast applications. Since digital signals have widely replaced analogue video signals, the YUV colour space has been superseded by the \( YC_bC_r \) colour space. This colour space is in principle identical to the YUV colour space but uses different conversion coefficients to accommodate for digitally reproducing the analogue signal [Poynton, 2003]. In the \( YC_bC_r \) colour space, the Y component is the luminance, the \( C_b \) component is the difference to the blue colour component, and the \( C_r \) is the difference to the red colour component. This colour space is the native colour space for most compressed video encoding standards, such as MPEG-1 and MPEG-2. The following equations can be used to convert \( YC_bC_r \) colour into RGB colour [Poynton, 2003]:

\[
\begin{align*}
R &= Y + 0.000C_b + 1.403C_r \\
G &= Y - 0.344C_b - 0.714C_r \\
B &= Y + 1.773C_b + 0.000C_r
\end{align*}
\]  

(2.2)

The HSV colour model, a derivative of the RGB colour model, represents colours by the three components hue, saturation, and value [Fairchild, 2005]. This model is sometimes referred to as the HSB (hue, saturation, brightness) colour model, and is commonly used in computer graphics applications since it is perceptually more uniform than the RGB and many other models [Poynton, 2003]. A benefit of the HSV colour model that is of importance to our work is that it separates the colour type (hue) from its saturation and its brightness. This is useful to reduce unwanted effects of sudden changes in the lighting conditions in video scenes. As illustrated in Figure 2.2, a colour space using the HSV model is usually represented as a cone because the Hue (H) is given in degrees such that \( H \in \{0, \ldots , 360\} \). The Saturation (S) and the Value (V) are specified in the range of \( \{0, \ldots , 1\} \). Assuming normalised RGB colour values so that \( R, G, B \in \{0, \ldots , 1\} \), RGB colour can be converted to
Figure 2.2: The HSV colour space is usually represented as a cone in which the hue is given as degrees in the range of \( \{0, \ldots, 360\} \). Saturation and value are usually specified as values between 0 and 1. Image reproduced from Wikimedia Commons (http://commons.wikimedia.org/wiki/Image:HSV_cone.png) under the terms of the GNU Free Documentation License [(FSF), 2002].

HSV colour as follows [Foley et al., 1993]:

\[
H = \begin{cases} 
\text{undefined} & \text{if } \text{MAX}_{RGB} = \text{MIN}_{RGB} \\
60 \cdot \frac{G-B}{\text{MAX}_{RGB} - \text{MIN}_{RGB}} & \text{if } \text{MAX}_{RGB} = \text{R} \text{ and } G \geq B \\
60 \cdot \frac{G-B}{\text{MAX}_{RGB} - \text{MIN}_{RGB}} + 360 & \text{if } \text{MAX}_{RGB} = \text{R} \text{ and } G < B \\
60 \cdot \frac{R-G}{\text{MAX}_{RGB} - \text{MIN}_{RGB}} + 120 & \text{if } \text{MAX}_{RGB} = \text{G} \\
60 \cdot \frac{B-R}{\text{MAX}_{RGB} - \text{MIN}_{RGB}} + 240 & \text{if } \text{MAX}_{RGB} = \text{B}
\end{cases}
\]

\[
S = \begin{cases} 
0 & \text{if } \text{MAX}_{RGB} = 0 \\
\frac{\text{MAX}_{RGB} - \text{MIN}_{RGB}}{\text{MAX}_{RGB}} & \text{otherwise}
\end{cases}
\]

\[
V = \text{MAX}_{RGB}
\]

In these equations, \( \text{MAX}_{RGB} \) is the maximum of the \((R,G,B)\) values and \( \text{MIN}_{RGB} \) is the minimum of the \((R,G,B)\) values. For practical purposes, the hue may be set to \( H = 0 \) if \( \text{MAX}_{RGB} = \text{MIN}_{RGB} \). HSV colour can then be generated from \( YC_bC_r \) colour by applying the transformation to RGB, as shown in Equation 2.2, and subsequently transforming RGB colour to HSV as shown in Equation 2.3.
2.1.3 Feature Representation

One of the key requirements for video databases to be useful is the provision of an operation that can compare video sequences or parts thereof. While it is trivial to compare digitally represented data for absolute equality, this is usually not desired. Video retrieval rather necessitates similarity-based techniques that enable retrieval of data by matching meanings [Santini and Jain, 1995; 1997]. For example, a television broadcaster might want to retrieve all video shots of the past year that show a soccer player scoring a goal. While the shots that satisfy this information need share the same meaning, they will certainly not be identical.

To enable similarity-based search, we must choose a representation of the content that allows for matching only the meaningful features for the given information need. Such features can be colour, shape, texture, sound, motion, or any other generic information that can be extracted from the content. For our example of finding all shots showing a soccer goal, we could try to identify the colour and texture of the soccer green, the shape of the goalposts, the shape and movements of one or more players on the green, or the soundscape in the stadium when a goal is scored. As such features by themselves have only little semantic meaning, we refer to them as low-level features [Chang and Smith, 1995]. Low-level feature extraction and similarity computation between images, frames, and video segments are key concepts of video and image retrieval. We discuss the most frequently used low-level features below, focusing on histogram features because these are in widespread use and are most relevant to our work.

Colour histograms can be created from images by partitioning the colour space that usually contains a large number of colours into a smaller number of colours. A colour histogram thus represents the distribution of colours in the image. Let us consider an example of an image \( I[X,Y] \) of width \( X \) and height \( Y \) in the HSV colour space with \( K \) possible colours, forming the codebook \( C = \{ c_0, \ldots, c_k, \ldots, c_{K-1} \} \). This means that \( K \) is the gamut of this colour space. For each colour described by the three components \(( H, S, V )\) there exists exactly one codebook entry \( c_k \). We then define \( M \) bins, with \( M < K \), to map the \( K \) initial colours to the smaller set of \( M \) colours, resulting in the codebook of target colours \( C' = \{ c'_0, \ldots, c'_m, \ldots, c'_{M-1} \} \). The value of \( I_{xy} \) is the index of the codebook entry in \( C \), describing the colour of the pixel at position \(( x, y )\) in the image \( I[X,Y] \). We define a quantiser function \( Q \) that maps the codebook \( C \) containing the initial \( K \) colours in the three-dimensional
2.1. STRUCTURE AND REPRESENTATION OF DIGITAL VIDEO

Figure 2.3: An example video frame of a television news program (left) and its three-dimensional colour histogram in the HSV colour space (right). As can be seen, the colour distribution is represented as differently sized spheres in the three-dimensional colour space. This histogram was visualised with the ImageJ [Rasband, 2006] 3D Colour Inspector Plugin [Barthel, 2006].

space to M bins in the codebook $C'$ as follows:\footnote{We use $\lfloor x \rfloor$ to denote the “floor” function to retain the largest integer less than or equal to $x$.}

$$Q(I_{xy}) = \left\lfloor \frac{I_{xy}M}{K} \right\rfloor$$

(2.4)

The colour histogram $h[m]$ is extracted by counting the number of pixels in the image that fall into each bin $m \in \{0, 1, \ldots, M - 1\}$ [Smith, 2001]:

$$h[m] = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \begin{cases} 1 & \text{if } Q(I_{xy}) = m \\ 0 & \text{otherwise} \end{cases}$$

(2.5)

This process results in a colour histogram in the three-dimensional space, also referred to as a 3D colour histogram as illustrated in Figure 2.3. This figure shows the 3D colour histogram for an example video frame in the HSV colour space. As can be seen from this figure, the colour distribution is represented as differently sized spheres in the three-dimensional colour space. Each sphere is one bin $m$; the size of the sphere represents the number of pixels $h[m]$ from the image that fall into this bin. In the histogram of Figure 2.3, we can clearly identify the few blue pixels of the logo that occupy three bins, the larger number of
pixels of the tie occupying two bins, and the pixels of the background and suit as the largest spheres. Depending on the bin size that is chosen for a 3D colour histogram, this can result in relatively large amounts of data. Simpler representations may be generated by processing each colour component separately. For example, by only processing the V component for each pixel in the HSV colour space, a grey-level histogram can be generated. Another option is to quantise each component separately and then concatenate the bins of each component in one vector, which is the representation that we use, as discussed in Chapter 3. We refer to all such histogram representations as global histograms because we generate one histogram that includes the pixels from the entire frame.

An advantage of global colour histograms is that they are invariant to translation and rotation of the image [Swain and Ballard, 1990] because they do not preserve any spatial information. In addition, we can normalise histograms to make them robust to scaling. However, ignoring spatial information also causes a major disadvantage because we can no longer distinguish frames with differences caused by an object or a person moving in the scene. For example, a dark frame with a yellow blob in the upper right corner has the same histogram as a dark frame with a yellow blob in the lower right corner and we are unable to distinguish these frames using above histogram representation. To alleviate this, a common approach is to generate localised histograms [Smith, 2001] by dividing each frame into several regions, and by extracting a histogram for each region. Figure 2.4 illustrates this process using 4 × 4 equal sized regions for a frame so that it can be represented by sixteen histograms instead of one. This opens the possibility to use more sophisticated frame-comparisons, such as evaluating the similarities between corresponding regions.

Pass and Zabih [1996] proposed a refined histogram representation that they call the colour coherence vector. This representation addresses some of the shortcomings of colour histograms by considering the spatial colour coherence of pixels. The first steps of creating a colour coherence vector are to apply a blurring filter to the image and then to generate a global colour histogram as described above. The pixels in each histogram bin are then further partitioned based on spatial coherence. Each pixel is considered either spatially coherent or spatially incoherent by evaluating neighbouring pixels. A pixel is coherent if there are sufficiently many neighbouring pixels within the same bin, otherwise it is considered incoherent. Both the number of coherent and incoherent pixels are stored as a coherence pair for each bin.

Huang et al. [1997] describe an image feature representation that captures the spatial correlation of colours in the image. First, the colour space is quantised into a finite set of
2.1. STRUCTURE AND REPRESENTATION OF DIGITAL VIDEO

Figure 2.4: We generate localised histograms by dividing each frame into regions and extracting one histogram for each region. In this example, we use sixteen equal-sized regions.

colours. Second, for each pixel of a given colour, the probability of a pixel at the pre-defined distance $d$ having the same colour is computed. Huang et al. [1997] refer to this as an autocorrelogram, while a more complex representation considering any colour combination is termed colour correlogram. Because of the increased size of the colour correlogram feature vector the autocorrelogram is often preferred [Ma and Zhang, 1998]. Compared to colour histograms and colour coherence vectors, the autocorrelogram feature has been reported to have better similarity matching performance in image retrieval experiments [Huang et al., 1997; Ma and Zhang, 1998].

Rautiainen and Doermann [2002] propose an extension to autocorrelograms that they name the temporal colour correlogram. This considers the changes in the colour distribution across several frames in a video; the authors report improved results when this feature is compared to autocorrelograms and colour histograms in video search experiments. As multiple frames must be processed for each video segment during the feature extraction step, this is computationally more expensive compared to purely spatial features.

A very compact representation of the colour distribution in an image can be attained by using colour moments [Stricker and Orengo, 1995]. In particular, the first three low-order moments across all pixels of the image are extracted, representing the average $\mu$, the variance $\sigma$, and the skewedness $\theta$. Considering again an image $\tilde{f}[X,Y]$ with $X \times Y$ pixels in the HSV colour space, the first three colour moments for one colour component, for example $H$, can
be obtained as follows [Acharya and Ray, 2005]:

\[
\mu_H = \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} I_{xy}^H
\]  

(2.6)

\[
\sigma_H = \left[ \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (I_{xy}^H - \mu_h)^2 \right]^{\frac{1}{2}}
\]  

(2.7)

\[
\theta_H = \left[ \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (I_{xy}^H - \mu_h)^3 \right]^{\frac{1}{3}}
\]  

(2.8)

In these equations, \( I_{xy}^H \) denotes the codebook value of the H colour component of the pixel at the location \((x, y)\) of the image. This is done for each colour component, which results in a feature vector with only nine elements to represent an image in a three-dimensional colour space.

All the features that we have discussed so far have been concerned with colour and its distribution across the image. However, the human visual system, besides being able to distinguish colours, relies also on other visual cues, such as shape and texture. Edge and shape information is frequently used as a feature in video retrieval. As for the colour-based features, these are usually generated for each frame and represented mathematically as a multi-dimensional vector. The principle of detecting edges is to identify local changes in the colour intensity level that lies above a set threshold. The Sobel [1968] edge detection algorithm [Duda and Hart, 1973] and the Canny [1986] edge detector are most frequently used, and we refer to the relevant literature [Acharya and Ray, 2005; Bovik, 2000; Castleman, 1996; Nixon and Aguado, 2002] for a detailed description of other approaches. Figure 2.5 shows our example video frame and the result of applying a Sobel edge detection filter to it. The Canny edge detection algorithm is based on the Sobel algorithm but applies additional steps which result in thinner edges that appear more pleasing to human viewers. However, the Sobel algorithm is often preferred in image and video processing as it is computationally less expensive and in most cases sufficient. Edge detectors can also output edge information that includes the edge type as one of the five classes: horizontal, vertical, diagonal (one for each direction), and non-directional. This information may be used to generate an edge histogram descriptor [Won, 2004; Won et al., 2002] for an image. Another application of edge detection is spatial segmentation of images and video frames. This is useful for decomposing an image into regions defined by shapes, primarily with the aim of separating objects in the foreground
2.1. STRUCTURE AND REPRESENTATION OF DIGITAL VIDEO

Figure 2.5: An example video frame of a television news program before (left) and after applying a Sobel edge detection filter to it (right). Shapes of objects and people can be identified well with this representation.

from the background, or extracting regions of similar colour or texture [Castleman, 1996]. However, spatial segmentation is not common in video retrieval because of its computational complexity [Grosky et al., 1997; Petković and Jonker, 2004] and we do not discuss it here.

Texture is the third important visual feature for humans besides colour and shape. The human visual system uses textures to identify objects, certain materials, and surfaces [Chang and Smith, 1995] such as grass, hair, clouds, fabric, the pavement of roads, and bricks. Automatic detection and classification of texture is an actively researched area and has been widely used for analysis of aerial, satellite, and medical images [Ma and Zhang, 1998; Manjunath et al., 2000]. The techniques that are used can broadly be categorised into statistical and structural approaches [Nixon and Aguado, 2002]. Most common among the statistical approaches is the co-occurrence matrix [Aksoy and Haralick; Haralick et al., 1973] which is based on an evaluation of the grey-level pixel intensities of an image under consideration of the distances at which identical pixel intensities co-occur. Structural approaches exploit the fact that most textures are composed of repetitively arranged texture primitives [Castleman, 1996]. Accordingly, structural approaches often use some form of spatial-frequency transformation, such as the Fourier and the wavelet transforms [Smith, 2001]. In particular, Gabor wavelets are widely used for this purpose and are reported to allow effective rotation-invariant texture classification [Haley and Manjunath, 1999; Manjunath et al., 2000].
There are non-visual features that are important for video retrieval. Since a video stream usually contains an audio track alongside the visual track, we can utilise the sound information of the audio track to extract useful information on the video content. In contrast to visual features that mostly originate from the image retrieval domain, the use of sound features for video retrieval has not been as extensively researched. The only exception is the application of Automatic Speech Recognition (ASR) techniques to the video sound track to transcribe any spoken information into text. This method was first applied to radio news broadcasts and is known as spoken document retrieval [Garofolo et al., 1999]. Employing speech recognition techniques has probably supplied research in content-based video retrieval with the most significant advancements [Hauptmann, 2005] and we discuss speech-based video retrieval in detail in Section 2.4.1.

The usefulness of the raw audio signal for video retrieval is limited by frequently occurring sections with silence or music overlay. Srinivasan et al. [2005] provide an overview of low-level audio features that may be used in video retrieval. Their application is often concerned with a confined task or domain, such as the detection of physical violence, indexing specific events, or speaker change- and monologue detection [Adams et al., 2001; Baillie and Jose, 2004; Lu and Zhang, 2002; Moncrieff et al., 2001; Nock et al., 2002; Nwe and Li, 2005]. As part of the Informedia Digital Video Library project, researchers at Carnegie Mellon University use the raw audio signal to detect silence as an indicator for story segmentation [Hauptmann and Witbrock, 1998]. They monitor amplitude, signal-to-noise ratio, and background noise to detect speaker changes, or changes in the location. The Discrete Cosine Transform (DCT) [Ahmed et al., 1974] and the Discrete Fourier Transform (DFT)\(^6\) are commonly applied to the decibel spectrum of the audio signal. The result is called the Cepstrum [Bogert et al., 1963] and holds information about changes in different bands of the decibel spectrum. This information can be used for audio compression, but also for speaker identification and automatic speech recognition, usually after it is aligned to the Mel-scale [Stevens et al., 1937]. The Mel-scale is a subjective scale of pitches that was designed on the basis of experiments with human listeners. This transformation yields Mel Frequency Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980; Zheng et al., 2001], sometimes referred to as cepstral vectors, that are used for automatic speech recognition [Davis and Mermelstein, 1980; Hauptmann, 1995], but may also used directly as a low-level feature for audio classification [Baillie and Jose, 2004; Iyengar et al., 2002; Lu and Zhang, 2002; Nock et al., 2002]. Low-level audio fea-

\(^6\)The Fourier Transform is a widely used technique for signal processing. The most frequently used computer implementation is the Fast Fourier Transform (FFT) by Cooley and Tukey [1965].
2.1. STRUCTURE AND REPRESENTATION OF DIGITAL VIDEO

Features are increasingly used in combination with machine learning techniques, such as hidden Markov modelling and Gaussian mixture modelling, for content-based retrieval of audiovisual data [Liu et al., 1998; Scheirer and Slaney, 1997; Zhang and Kuo, 1998]. Adams et al. [2003] propose the use of several low-level audio features within a generalised machine learning framework for semantic video classification, however, they do not elaborate on details of these features. We discuss the application of machine learning techniques to video retrieval in more detail in Section 2.3.

An interesting aspect of these audio features is that they extend into the temporal domain and tend to better reflect events and activities than visual features that capture only the spatial arrangement of colours at a specific time. However, attempts to leverage information from visual features in the temporal domain do exist and usually capture differences between frames in some form. Accordingly, many of these techniques employ features from compressed video streams, such as the DCT coefficients or the motion vectors. Such features have been reported to be useful for detecting abrupt transitions in video [Arman et al., 1994; Heng et al., 1999; Nang et al., 1999; Patel and Sethi, 1996] because these usually come in tandem with significant changes within few frames. A good overview of such techniques is given by Mandal et al. [1999]. A temporal feature that is applied to uncompressed video is the temporal colour correlogram proposed by Rautiainen and Doermann [2002] that we have described on page 20. Similarly, DeMenthon and Doermann [2003] propose spatio-temporal descriptors that include the motion of pixels across multiple frames as well as their colour. They report that these descriptors are useful for near-duplicate detection in video, but emphasise that the complex computation makes real-time query processing currently not feasible.

2.1.4 Distance Metrics

Once low-level features have been extracted, the next question is how to use these to compare video frames or video shots with each other. Such an operation is closely tied to the way these features are represented. Low-level features such as those that we have discussed are stored as high-dimensional vectors, and the standard way to perform similarity matching is therefore to use some geometric distance metric. Under the assumption of a Minkowski space [Naber, 1992], let \( \vec{a} \) and \( \vec{b} \) be two feature vectors of dimension \( K \). The generalised Minkowski distance metric is then specified as [Androutsos et al., 1998]:

\[
L_M(\vec{a}, \vec{b}) = \left( \sum_{k=0}^{K-1} |a_k - b_k|^M \right)^{\frac{1}{M}}
\]  

(2.9)
CHAPTER 2. FUNDAMENTAL ASPECTS OF CONTENT-BASED VIDEO RETRIEVAL

Figure 2.6: The $L_M$ norm distances in comparison for $M \in \{1, 2, \infty\}$ in a two-dimensional space. The special cases for $M = 1$ and $M = 2$ are most commonly used in information retrieval.

where $a_k$ and $b_k$ are the $k$th element of the vectors $\vec{a}$ and $\vec{b}$. This is also referred to as the $L_M$ norm; $M$ denotes the order of the norm. The special cases for $M = 1$ and $M = 2$ are most commonly used. The $L_1$ distance ($L_1$ norm) is better known as the Manhattan distance, or city-block distance, because it equals the shortest distance between two points if one is restricted to travel in square blocks. This is, for example, the case when commuting in a car in Manhattan.\(^7\) The $L_2$ distance is also known as the Euclidean distance and represents the shortest distance between two points if it is measured directly. An illustrated comparison of these low-order $L_M$ norm distances in a two-dimensional space is shown in Figure 2.6, using two points on a chessboard. This figure also shows the supremum norm $L_\infty$, known as the Chebyshev distance. The Chebyshev distance is defined as the distance between two points in a vector space that is the greatest of the distances along any coordinate dimension [Abello et al., 2002]. From Equation 2.9, with $M = \infty$ follows that:

$$L_\infty(\vec{a}, \vec{b}) = \max_k(|a_k - b_k|) = \lim_{M \to \infty} \left( \sum_{k=0}^{K-1} |a_k - b_k|^M \right)^{\frac{1}{M}}$$ \hspace{1cm} (2.10)

As can be seen from Figure 2.6, the Chebyshev distance is always less than or equal to the Manhattan distance.

\(^7\)The $L_1$ distance is therefore sometimes called “Taxi-cab distance”.

25
From the laws of trigonometry, we derive the well-known angular distance measure that is also referred to as the Cosine measure. This measure is widely used to compute rankings in text search engines [Zobel and Moffat, 2006]. It has also been applied to image and video retrieval [Androutsos et al., 1998] and is defined as follows [Androutsos et al., 1998]:

$$d_\theta = 1 - \frac{2}{\pi} \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||}\right)$$

(2.11)

In image retrieval experiments conducted by Androutsos et al. [1998] comparing the $L_M$ norm distances, the angular distance, and several other vector distance measures, the angular distance measure outperformed all other distance measures that were used. However, in their experiments they only considered an image representation using global colour histograms in the RGB colour space. Conversely, Ma and Zhang [1998] report strong results in video segmentation experiments when using global HSV colour histograms and colour coherence vectors in combination with the $L_1$ and $L_2$ distance measures. This shows that feature representation and distance measure must be harmonised, and that the performance of both depends much on their suitability for the given task.

Swain and Ballard [1991] propose a distance metric called Histogram Intersection that is designed for comparing colour histograms. Specifically, they report this measure to be less influenced by pixels in the background of a frame. Given two histogram vectors $\vec{a}$ and $\vec{b}$ with $K$ dimensions, Swain and Ballard [1991] define the histogram intersection as follows:

$$d_I(\vec{a}, \vec{b}) = \sum_{k=0}^{K-1} \frac{\min(a[k], b[k])}{\sum_{k=0}^{K-1} b[k]}$$

(2.12)

Swain and Ballard [1991] note that this measure is identical to the $L_1$ norm if the compared histograms are normalised, for example, when histogram bin counts are expressed as percentages. Normalisation is equivalent to scaling frames to identical sizes prior to histogram extraction. This step can normally be omitted when comparing frames within one video clip. A statistical approach to compute frame differences uses the well known $\chi^2$ test. Nagasaka and Tanaka [1992] proposed to use this test to assess the difference of grey-level histograms:

$$d_{\chi^2}(\vec{a}, \vec{b}) = \sum_{k=0}^{K-1} \frac{(a[k], b[k])^2}{b[k]}$$

(2.13)

This measure is reported to better reflect the differences between two frames [Nagasaka and Tanaka, 1992] when performing shot transition detection; it can be seen as a measure of how unlikely it is that the colour distribution of the first frame is represented by the
colour distribution of the second frame. However, Zhang et al. [1993] could not confirm better performance with the $\chi^2$ test compared to the $L_1$ norm in their transition detection experiments, and note the increased computational complexity of the $\chi^2$ test.

Besides these generic geometric distance metrics, various distance measures for entire video sequences have been proposed [Adjeroh et al., 1999; Cheung and Zakhor, 2003; Lee et al., 2000; Rubner et al., 2000]. These can be applied to video shots and take the changes between frames over time into account. Due to their computational complexity, they are not scalable to retrieval in large collections [DeMenthon and Doermann, 2003; Rubner et al., 2000], and usually designed for specific purposes, such as video sequence matching [Adjeroh et al., 1999] or near-duplicate identification for entire video clips [Peng and Ngo, 2004]. Moreover, most of these distance measures are inseparably tied to a specialised low-level feature extraction method.

A common strategy in video retrieval is to reduce complexity to provide more scalable methods that can be applied to large video collections. This can be done by segmenting video into smaller semantic entities, usually shots, and by selecting one frame to serve as a representative image for each shot [Idris and Panchanathan, 1997]. The video retrieval problem can thus be largely reduced to an image retrieval problem, and the low-level feature representations and geometric distance metrics that we have discussed can be applied. The necessary step for this reduction is to segment video within the temporal domain and we discuss this step in the next section.

### 2.2 Temporal Segmentation

Given that video shots usually contain confined semantics concerning a specific location, person, or setting [Konigsberg, 1989], shots are commonly chosen as the basic unit for storage and retrieval in video databases. A video shot may be represented by a single image, also called the key frame, that contains enough visual information to sufficiently convey the content of the full shot [Brunelli et al., 1999]. This approximation can be used when visualising results, and also when classifying video shots based on their visual content. Moreover, after extracting key frames, a system may permit users to use an example image as a query and, using content-based image retrieval techniques, return relevant video shots on the basis of the visual content of their key frames. As shots can be seen as basic building blocks of a video, they may also be combined to form stories [Del Bimbo, 2001] that cover a specific topic comprehensively. This can be useful to enable video retrieval beyond the granularity
of shots [Hauptmann and Witbrock, 1998; Merlino et al., 1997; Yeung and Yeo, 1996; Yeung et al., 1996].

2.2.1 Review of Existing Techniques

The most common approach to video shot segmentation is identifying the transitions that have been inserted when concatenating the individual shots. We distinguish two main classes of transitions that mark the boundaries between consecutive shots. Abrupt transitions or cuts are the simplest and most common transition type. A cut is a shot change between two adjacent frames with no overlap between the shots. In contrast, gradual transitions extend over several frames such that the frame content of the new shot gradually replaces the content of the previous shot. This is usually achieved by applying editing effects such as fades, dissolves, wipes, and other spatial edits. As a consequence, gradual transitions are more complex, and thus more difficult to detect than cuts [Aigrain et al., 1996; Smeaton et al., 2003; Zhang et al., 1993]. Early work on shot boundary detection has focused on detecting cuts [Nagasaka and Tanaka, 1992; Ueda et al., 1991] and neglected gradual transitions. However, with the availability of digital video editing systems, gradual transitions have become more common in modern footage. This is reflected by the TRECVID shot boundary test collections that we use in our evaluation. We describe the TRECVID workshop in detail below and present statistics on the test collections in Appendix A. From these statistics, we can see that approximately 65% of all transitions in the 2001 test set are cuts, while roughly 33% are dissolves and fades. Conversely, in the 2006 test collection, consisting entirely of modern television footage, only 49% of all transitions are cuts and 51% are dissolves, fades, and other effects. Lienhart [1998] reports that together, cuts, fades, and dissolves account for approximately 99% of all transitions in all types of video. Fades and dissolves are the most common forms of transition besides cuts, and their accurate identification is important for effective video retrieval.

As previously described, modern footage is usually encoded using formats such as MPEG-1 or MPEG-2. Shot boundary detection can be performed by evaluating DCT coefficients and motion vectors directly from the compressed video stream. We can thus categorise shot boundary detection techniques as using either compressed video or uncompressed video. Techniques using compressed footage have the potential to be very efficient because the video stream does not need to be fully decoded [Jun et al., 2000; Meng et al., 1995; Sugano et al., 2003; Xiong and Lee, 1998; Yeo and Liu, 1995]. However, using the encoded features
directly tends to result in lower accuracy [Boreczky and Rowe, 1996; Koprinska and Carrato, 2001] than approaches to shot boundary detection using uncompressed video and many of the approaches in the compressed domain do not consider gradual transitions [Heng et al., 1999; Nang et al., 1999; Patel and Sethi, 1996]. Moreover, techniques for uncompressed video are independent of the compression scheme. More detailed overviews of techniques using encoded video are given by Mandal et al. [1999], Brunelli et al. [1999], and Koprinska and Carrato [2001]. We do not discuss such methods in detail here, and focus only on shot boundary detection schemes that process uncompressed video.

Most techniques for shot boundary detection are based on the assumption that frames within a shot tend to be similar. A common approach is therefore to compare adjacent frames, and detect transitions by applying a threshold to the inter-frame difference. Nagasaka and Tanaka [1992] propose to extract grey-level histograms from each frame and compute frame differences with the $\chi^2$ test. The $\chi^2$ test is known to have an enhancing effect on frame differences [Nagasaka and Tanaka, 1992; Zhang et al., 1993] which can benefit cut detection, but it also increases the sensitivity to camera motion and rapid object movements. To reduce this sensitivity, Nagasaka and Tanaka [1992] divide each frame into $4 \times 4$ equal sized regions — sub-frames — and compare corresponding frame regions instead of the full frames. After discarding the eight largest sub-frame differences, they use the sum of the remaining eight sub-frame differences for inter-frame comparison. Zhang et al. [1993] have reported this method to be computationally more expensive and not to necessarily yield better results than the much simpler Manhattan distance measure. Shahraray [1995] proposes to divide each frame into twelve blocks and to compute differences of corresponding blocks on a pixel-by-pixel basis, using their intensity values. An Image Match (IM) value is then calculated by applying a nonlinear digital order statistic filter that considers the order of the individual block match values and is robust against local inter-frame differences.

These early proposals seem more suitable for cut detection as they compare only adjacent frames. To also detect gradual transitions, Shahraray [1995] uses a different evaluation scheme to identify small, continuous changes over several frames when monitoring the IM signal. Zhang et al. [1993] present the twin-comparison approach that uses two thresholds. Once a first, lower threshold is exceeded by the inter-frame difference, the current frame is identified as the possible start of a gradual transition. Subsequent frames are then compared against this frame to accumulate the inter-frame distance. In addition, consecutive frames are compared against each other during a possible gradual transition. The end of the gradual transition is identified if the difference between consecutive frames is lower than the first,
lower threshold, and the accumulated inter-frame difference is larger than the second, higher threshold. However, with these approaches it is difficult to distinguish between subtle frame differences caused by normal scene activity and frame differences caused by gradual transitions [Shahraray, 1995; Zhang et al., 1993]. Additional motion estimation is necessary to reduce false detections due to rapid scene activity [Shahraray, 1995; Zhang et al., 1993].

Most of this work also lacks large-scale performance evaluation. While experimental results are presented in some cases [Hampapur et al., 1994; Zhang et al., 1993], the test sets are usually small (much less than one hour of video) and no standardised evaluation methods have been used. A well-defined evaluation of different shot boundary detection strategies is presented by Boreczky and Rowe [1996], using a test set of approximately 3 hours and 50 minutes in duration. This test set comprises movies, television news, a cartoon, and television advertisements with a total of 2507 cuts and 506 gradual transitions. Boreczky and Rowe [1996] conclude from their results that a relatively simple histogram-based approach using frame regions performs best. They attest that gradual transition detection still poses a problem, while abrupt transitions can be detected relatively well with the tested methods.

More recent research focuses on gradual transition detection [Heesch et al., 2003; Sun et al., 2001; Yu and Srinath, 2001; Zhang et al., 2001] by evaluating variations of the twin-comparison algorithm first proposed by Zhang et al. [1993]. Pickering and Rüger [2001] divide frames into nine blocks, and extract the red, green, and blue colour component histograms of each. The distance between the histograms of corresponding blocks is calculated, and the largest of the three is retained. The median of the nine individual inter-block distances is taken as the inter-frame distance. A transition is reported if this distance is greater than a fixed threshold and also greater than the average of the distance values for the 32 surrounding frames. This algorithm is reported to be sensitive to camera and object motion, and is also limited by the use of constant thresholds.

Many techniques combine several features such as colour histograms, edge detection [Lienhart, 2001a], motion estimation [Quénot and Mulhem, 1999], spatial features [Naphade et al., 1998b], and texture [Adams et al., 2002]. While such methods can improve the accuracy of both cut and gradual transition detection [Quénot et al., 2003], they increase the computational complexity of the algorithms. Some video segmentation systems also consider features such as audio information or captions [Hauptmann and Witbrock, 1998; Pfeiffer et al., 1998]. These are usually designed for a particular task on specific types of content, for example the detection of television advertisements or identifying breaks in video conferencing [Lienhart, 1998].
Some approaches have been proposed that are based on the video production model, such as the work of Hampapur et al. [1994; 1995]. In this work, the authors define edit models for four different types of transitions based on the operation of video editing systems. Transitions are detected by monitoring one or more features of the video for patterns very similar to those predicted by the internal models. More recently, [Lienhart, 2001a;b], Liu and Chen [2002], and Nam and Tewfik [2000] have proposed similar techniques that use different variations of features during the detection phase. While these approaches have shown promising results, they can only work effectively if all transitions that exist in the test collection can be classified with one of the pre-defined transition types. Moreover, we are not aware of any large-scale evaluation of these approaches that use realistic datasets. This is unfortunately the case with most of the approaches that we have discussed so far, mostly due to a lack of standardised test collections and a common evaluation methodology.

This situation has improved with the introduction of the TREC Video Retrieval Evaluation (TRECVID)\(^8\) in 2001. Successful approaches presented at TRECVID usually combine several features and techniques to perform shot boundary detection. The IBM CueVideo\(^9\) program extracts sampled three-dimensional colour histograms from video frames [Smith, 2001], it uses a moving window of frames to compute statistics of frame differences and adaptive threshold levels. A state machine is then applied to detect and classify transitions. A later version of this system uses also edge gradient histograms [Adams et al., 2002; Amir et al., 2003] and compares pairs of frames that are up to seven frames apart.

The shot boundary detection system by CLIPS-IMAG\(^10\) has shown to perform well using pixel-by-pixel comparison of frames with motion compensation and flash detection [Quénot et al., 2002; 2003]. A group from the Imperial College, London describes a system that uses the histogram difference within adjacent frames [Heesch et al., 2003; Pickering et al., 2002]. It employs the twin-comparison algorithm [Zhang et al., 1993] for gradual transition detection with empirically determined thresholds based on training runs using earlier TRECVID test collections. The Heinrich-Hertz-Institute in Germany, part of the Fraunhofer Institute for Telecommunications, have presented a system for shot boundary detection using luminance of sub-sampled pixel data, edge differences, flash detection, motion compensation, and the Hough [1962] transform of frames [Petersohn, 2004]. While this system has been shown to be very effective, it is also relatively efficient due to the sub-sampling of the frame data.

\(^8\)http://www-nlpir.nist.gov/projects/trecvid
\(^9\)http://alphaworks.ibm.com/tech/cuevideo
\(^10\)http://www-clips.imag.fr
2.2. TEMPORAL SEGMENTATION

Other approaches involve applying transforms to the frame data. Cooper et al. [2003] represent frames by their low-order DCT coefficients, and calculate the similarity of each frame to the frames surrounding it. Miene et al. [2003] use Fast Fourier Transform (FFT) [Cooley and Tukey, 1965] coefficients calculated from a grey-scale version of the frame for their comparisons. Both of these techniques have performed well when detecting cuts but resulted in poor detection performance for gradual transitions. Tahaghoghi et al. [2002] proposed a ranking technique in a moving window of frames for effective cut detection. This approach applies the concepts of Query-By-Example (QBE) and ranking to the video segmentation problem and has been shown to be very effective with a feature derived from the wavelet transform of the frame data [Tahaghoghi et al., 2002]. This feature is generated by computing the six-tap Daubechies wavelet transform coefficients from the native YCbCr colour data of a frame. Tahaghoghi et al. used the Mallat [1989] algorithm to compute this feature, after re-arranging the frame data to allow comparison of different-size frames. The feature is therefore called the Wavelet transform for Re-ordered data (RWav) and is relatively expensive to compute. With this feature, their system outperformed most other systems in cut detection that participated in TRECVID 2002, but produced poor results in gradual transition detection [Tahaghoghi et al., 2002].

This review shows that video shot boundary detection is a problem that has been extensively researched, but also that performing such a seemingly simple task as detecting transitions in video requires complex processing steps. As Smeaton et al. [2003] assert, achieving highly accurate results continues to be a challenge, and shot boundary detection is not yet a solved problem. However, the ranking approach by Tahaghoghi et al. [2002] is very interesting as it has shown to be effective with only one low-level feature, and is not as complex as most other approaches. As this approach forms the basis of some of our own research, we describe it in more detail in the following section, primarily to define terminology used in Chapter 3, where we describe our approach to shot boundary detection.

2.2.2 A Ranking Approach to Cut Detection

The ranking approach defines two sets of frames, the pre-frames and the post-frames that surround the current frame $f_c$. Together, the pre- and post-frames are a moving window that is centred on the current frame; we refer to the window as moving because it is used to sequentially consider each frame in the video as possibly bordering a cut. The pre-frames are the frames preceding the current frame, and the post-frames are those that follow it. The
number of pre- and post-frames is always equal. Tahaghoghi et al. [2002] refer to this as the *half-window size*.

While advancing through a video frame-by-frame, each frame is considered as the current frame $f_c$. The distance between $f_c$ and each frame within the pre- and post-frames is computed. These frames are then ordered by increasing distance from the current frame to achieve a ranking. Considering only the first $\frac{|C|}{2}$ top-ranked frames — which is equal to the half-window size — the number that are pre-frames is recorded; we refer to the number of pre-frames in the $\frac{|C|}{2}$ top-ranked frames as the *pre-frame count*. The change in the pre-frame count over adjacent frames is used to accurately detect cuts. If the value of the pre-frame count drops to zero (or close to zero), it is likely that a cut has occurred because all pre-frames will belong to one shot, while the post-frames belong to the next shot. The pre-frames will thus be dissimilar from the post-frames. At the same time, the current frame will be positioned on the last frame of the previous shot. This results in the post-frames all being ranked highly, while the pre-frames will be ranked very low. The pre-frame count will thus be zero or close to zero.

The ranking approach considers the results of computing the pre-frame count for several adjacent frames to improve cut detection reliability. This requires setting an upper threshold $U_c$ that the pre-frame count must reach prior to a cut. When this occurs, the algorithm tests whether the pre-frame count falls below a minimum threshold $L_c$ within a few frames. A cut is reported if the pre-frame count traverses both thresholds, and if the following two requirements are met. First, when adjacent frames span a cut, the value of the pre-frame count must fall from near $\frac{|C|}{2}$ to 0 within a few frames; this is captured by monitoring the pre-frame count slope for large negative values. Second, the pre- and post-frames spanning a cut must be reasonably different. Tahaghoghi et al. [2002] used a fixed threshold that was set to 25% of the maximum possible inter-frame difference to accomplish this.

This approach allows effective cut detection in different types of video. However, enhancements are desirable to replace the fixed threshold that is used, and to improve the effectiveness of this approach when using simpler colour histogram features. Most importantly, the ranking approach is less effective on gradual transitions, and further work is required to develop a gradual transition detection scheme that can be integrated with the ranking approach.
2.2. TEMPORAL SEGMENTATION

2.2.3 Systematic Performance Evaluation

The annual Text REtrieval Conference TREC [Voorhees and Harman, 2005][11] included a video evaluation track for the first time in 2001. Since 2003, the video track has been an independent evaluation and workshop using metrics-based evaluation. Participants are invited to test their approaches in different tasks, such as shot boundary detection, high-level feature detection, and search. We use the TRECVID shot boundary test collections to evaluate the performance of our approach described in Chapter 3.

The total duration of each of the test sets ranges between approximately five and eight hours. Detailed information on these test sets can be found in Appendix A. Each of these test collections comprises up to 22 video clips of educational, promotional, documentary, and television news footage. The shot boundary test reference data for the TRECVID test collections is manually created at NIST and specifies four different types of transitions that are as follows [Smeaton et al., 2003]:

**Cut:** The last frame of a shot is immediately followed by the first frame of the new shot without any overlap between shots.

**Fade-out/-in (FOI):** Here, one shot gradually fades into a monochrome frame (usually black or white) after which the new shot gradually fades in from the monochrome frame.

**Dissolve (DIS):** This is defined as the gradual merging of the end of one shot and the beginning of another, produced by the superimposition of a fade-out onto a fade-in. A dissolve is also sometimes referred to as a cross-fade.

**Other (OTH):** All transitions that do not fit into one of the above categories. This includes wipes, flips [Konigsberg, 1989], and other special effects.

Dissolves, fades, and other transitions are considered gradual transitions because they usually extend over several frames, while cuts are abrupt shot changes. The transition length of a cut is thus zero. However, in the TRECVID ground-truth data, the last frame of the previous shot and the first frame of the new shot are included into the ground-truth data for a cut, resulting in the effective transition length of two frames for cuts.

We measure detection performance by evaluating Recall $R$ and Precision $P$. We adopt the definitions of Ruiloba et al. [1999]:

\[
R = \frac{N_C}{N_C + N_D} \quad P = \frac{N_C}{N_C + N_I}
\]  

(2.14)

where $N_C$ is the number of transitions that were correctly found, $N_D$ is the number of deleted transitions, and $N_I$ is the number of inserted transitions. That is, we refer to false negatives as deleted transitions and to false positives as inserted transitions. The sum of $N_C$ and $N_D$ equals the total number of transitions in the reference data. In addition, we use a quality measure $Q$ as defined by Quénot and Mulhem [1999]:

\[
Q = \frac{N_C - N_I}{N_C + N_D}
\]

(2.15)

Combining Equations 2.14 and Equation 2.15, the Quality measure may also be expressed as follows:

\[
Q = \frac{R}{3} \left( 4 - \frac{1}{P} \right)
\]

(2.16)

This measure takes into account that inserted transitions may be filtered out in subsequent processing, and so penalises these less than deleted transitions. To evaluate the performance in correctly detecting start and end of gradual transitions, we use the frame-based recall $R_F$ and the frame-based precision $P_F$ that are also used at TRECVID:\footnote{http://www-nlpir.nist.gov/projects/t2002v/abmeasures.html}

\[
R_F = \frac{F_O}{F_R} \quad P_F = \frac{F_O}{F_D}
\]

(2.17)

$F_O$ denotes the number of frames over which the detected transition and the reference transition overlap. $F_R$ is the defined length (in frames) of the reference transition, and $F_D$ is the detected length.

We evaluate system performance using the ComparisonManager software\footnote{http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools} that is offered by NIST to compare detected transitions with the reference. Dissolves, Fades, and Other transitions are not distinguished in terms of the evaluation and are all considered Gradual Transitions (GRA). The evaluation thus distinguishes between gradual transitions and abrupt transitions. Gradual transitions can match only gradual transitions, and cuts can match only cuts. The definition makes one exception: gradual transitions that are shorter than six frames, also called short graduals. Short graduals are perceived as cuts by humans and
the evaluation allows for them to be matched against both detected gradual transitions and detected cuts.

To cater for differences in the frame numbering that may occur due to different video encoders and decoders, reference transitions are expanded by five frames on each side when performing cut matching.

2.3 Semantic Classification

Early video retrieval approaches such as the Virage video search engine [Hampapur et al., 1997] and VideoQ [Chang et al., 1998] have adopted the QBE paradigm that is known from content-based image retrieval systems. Image retrieval systems such as VisualSEEk [Smith and Chang, 1996] and PhotoBook [Pentland et al., 1994] allow users to select an example image and search the database for similar images. Another well-known system is QBIC [Flickner et al., 1995; Niblack et al., 1993] that extends this paradigm to video search. When performing video search with this technique, users select an example image that is compared to video shot key frames that were extracted in a previous temporal segmentation step. Video shots are considered relevant if their key frame has a high similarity to the query example image. This approach is clearly limited [Adams et al., 2002] because matching is only performed on the basis of spatial low-level features of the example images and key frames. The VideoQ system attempts to leverage information from the temporal domain by including object movement into the query definition [Chang et al., 1998], but this also increases the complexity of formulating the query. Moreover, the general limitation of the QBE strategy, that a query example must be somehow provided, remains; search without an example image is not supported. The QBIC system does allow concept-based search, but the concept terms are manually defined [Flickner et al., 1995], which drastically limits the scalability of this approach.

These limitations have created a need for video search engines to operate similarly to text-document search engines and allow search using textual queries [Naphade et al., 2002]. A crucial aspect is therefore to bridge the gap between low-level visual content and high-level semantics [Smith et al., 2003]. Computers can access and handle low-level features relatively well, but they cannot understand meaning of visual content. However, if we can provide a mapping between low-level features and high-level semantic concepts, we can filter, search, and retrieve video content based on its meaning to humans with the aim of implementing effective video search engines.
CHAPTER 2. FUNDAMENTAL ASPECTS OF CONTENT-BASED VIDEO RETRIEVAL

Figure 2.7: An example of fitting a decision function — here linear — in a two-dimensional feature space to separate positive examples (blue) from negative examples (red). Depending on the distribution and the choice of the features, false classifications may occur because the decision function cannot be perfectly fitted.

2.3.1 Automatic Classification

Many researchers propose machine learning approaches to solve the problem of mapping semantics to low-level features [Barnard and Forsyth, 2001; Iyengar and Lippman, 1998; Naphade et al., 1998a]. The general strategy underlying such approaches is to extract low-level features from pre-classified examples in order to model the relationship between the low-level features and the classification of the examples [Theodoridis and Koutroumbas, 2006]. The task is to learn how to identify a specific class of images on the basis of the low-level features. This is also referred to as supervised learning and is usually formulated as a binary classification problem. The examples have to be labelled as either positive or negative with regards to a class specified a priori. After the learning phase, the goal is to classify unseen data automatically. Machine learning usually applies Bayesian decision theory so that posterior class-membership probabilities are estimated for each unseen item [Theodoridis and Koutroumbas, 2006]. Based on the feature data of an unseen item, the probability is estimated for this item to belong to either the negative or the positive group with respect to the previously defined class.

In video retrieval, this technique is usually applied to temporally segmented video [Adams et al., 2002; Iyengar and Lippman, 1998; Naphade et al., 2002], where the unit of retrieval is the shot. Low-level features are extracted for each video shot, and the semantic categories
2.3. SEMANTIC CLASSIFICATION

that we wish to use for classification must be defined [Naphade et al., 2002]. These categories usually refer to some object or visual semantic concept, such as Car, Mountain, or Fire and are sometimes called high-level features. Example shots are then manually labelled as those that can be categorised with the given concept (positive) and those that cannot be categorised with the concept (negative). During the training phase, the supervised learning algorithm learns how to distinguish positive from negative examples based on the previously extracted low-level features. Although this process is often called “learning” or “training”, it is merely a statistical modelling process in which a function is fitted within the multidimensional input feature space that optimally divides positive and negative examples [Theodoridis and Koutroumbas, 2006]. This is shown in Figure 2.7 using a linear function to divide positive and negative examples in a two-dimensional feature space. However, for most real-world applications such as video retrieval, the feature space is of high dimensionality and the function to separate positive and negative examples defines a non-linear hyperplane [Theodoridis and Koutroumbas, 2006]. As the graph in Figure 2.7 illustrates, the decision function may not always be perfectly fitted without causing false classifications. This depends on the distribution of the examples in the feature space, the choice of features, and the choice of the decision function [Theodoridis and Koutroumbas, 2006].

The low-level features to use is subject of ongoing research and most recent results suggest that a combination of many features works best [Adams et al., 2003; Campbell et al., 2006; Cao et al., 2006; Snoek et al., 2006c]. Similarly important is the question of which technique to use for modelling the input data and fitting the decision function. Popular choices are Hidden Markov Models (HMM) [Baum et al., 1970] or Gaussian Mixture Models (GMM) [Titterington et al., 1986] that can be estimated with the well-known Expectation Maximisation (EM) algorithm [Dempster et al., 1977]. Naphade et al. [1998a; 2002] use HMMs and GMMs in their early work on semantic video classification. Similarly, Iyengar and Lippman [1998] use HMMs to automatically separate news video sequences from sports video sequences. More recently, Support Vector Machines (SVM) [Vapnik, 1995] have proven to yield good results for video shot classification tasks [Amir et al., 2005; Lin and Hauptmann, 2002; Smith et al., 2003; Snoek et al., 2006c]. Adams et al. [2002] compare Gaussian mixture models and support vector machines when applied in a generalised framework for multimedia indexing and conclude that the SVMs outperform GMMs in this task. Consistently, recent work in the field of video indexing and retrieval relies on using SVMs for automatic content-based classification and clustering tasks [Campbell et al., 2006; Cao et al., 2006; Snoek et al., 2006b,c]. A trend-setting conclusion from this work is that content-based
video retrieval can benefit from this model-based approach as long as very many semantic concepts can be trained [Hauptmann et al., 2007; Snoek et al., 2006a; 2007].

As Naphade et al. [2002] point out, a crucial requirement of supervised learning techniques is the availability of adequate training examples. These must be annotated manually by human reviewers, which is usually done at the shot-level [Adams et al., 2002]. The success of supervised learning techniques for video retrieval has thus come in tandem with an increased need for manually annotated training collections. This has only recently been identified as a research problem; Naphade and Kender [2004] show that the performance of supervised learning algorithms is heavily dependent on the number of training examples that are available. They report that “in general, the detection performance seems to be proportional to the logarithm of the number of positive training examples” [Naphade and Kender, 2004]. Quénot and Ayache [2005] report that they use 2048 positively labelled training examples for each concept and twice that number of negatively labelled examples. Snoek et al. [2006c] confirm this relationship between the number of training examples and retrieval performance in large-scale experiments using 101 semantic concepts that are trained with up to 33,000 examples. They report that reasonable performance could be achieved as long as at least 5% of their training examples were labelled positive for a given concept. This equates to approximately 2,200 positive training examples, given that they used 70% of the TRECVID 2005 development corpus for training which consists of 61,904 annotated shots. However, their results suggest that better performance can be achieved if the number of training examples is much larger than 2,200. While some researchers address the problem of training semantic concept detectors with few examples [Juan et al., 2006; Natsev et al., 2005] and report promising results, the proposed approaches can only attenuate the problem. Snoek et al. [2006a] conclude that training robust concept detectors with few examples remains problematic.

As we describe below, besides requiring a large number of annotated examples, supervised learning algorithms perform best with training examples that are clearly and consistently classified using a binary classification scheme. Visual information, however, may be interpreted very individually and unambiguous classification is often — even for humans — a difficult task. It is therefore not only of importance to provide very many annotated examples but also to provide consistently and unambiguously classified examples.
2.3. SEMANTIC CLASSIFICATION

2.3.2 Manual Annotation

The trend to share information on the Web, especially in web-logs and photo albums, has led to many large annotated image collections. Photo sharing applications such as Flickr\(^{14}\) allow users to assign free-text labels to each image. Researchers at Carnegie Mellon University harness the community effort in their ESP Game [von Ahn and Dabbish, 2004] that implements manual image annotation as a game-like application. Users are asked to label images that are found by crawling the Internet. The system combines multiple annotations for each image, and computes a confidence score based on how many different users agree on a particular description of that image. Users in turn collect points if they agree with the majority opinion. Google has adopted this scheme for their Image Labeler feature.\(^{15}\) Two users are randomly paired to annotate the same set of images over a 90-second period with as many free-text labels as possible. Users earn points for each matching label between them.

While these approaches are very interesting, the usefulness of annotated images from such collections for training semantic concept classifiers has not been shown. The annotations are highly subjective and do not use a controlled concept lexicon. For example, using the search term “aircraft” on Flickr retrieves many images of people sitting in passenger aircraft or aerial views out of aeroplanes. But few of the returned images actually depict aeroplanes. The annotation is often based on individual thoughts or experiences that users relate to the images, but they do not necessarily reflect the image content well. While there is no “right” or “wrong” when it comes to the interpretation of visual content, annotation in such an uncontrolled environment tends to be very noisy. This noise renders such collections less suitable for learning specific concepts with supervised learning algorithms. Moreover, it is likely that models that are trained from such diverse photo collections result in poor classification performance for video retrieval because of the differences in quality, production style, and lighting conditions. To generate collections for effective training of supervised learning algorithms, we require the annotation to be conducted in a more controlled environment. Most importantly, the vocabulary of concept terms should be pre-defined and limited, and the collection that is used should relate to the target domain. That is, if we wish to perform automatic detection of television news footage, we should ideally use television news video for training.

\(^{14}\)http://www.flickr.com
\(^{15}\)http://images.google.com/imagelabeler
To address this issue and generate an adequately annotated training collection for video retrieval, the TRECVID annotation forum was conducted in 2003 [Lin et al., 2003]. This annotation forum is an initiative by many of the regular TREC Video Retrieval Evaluation [Over et al., 2005; Smeaton et al., 2003] participants and aimed at providing a common training collection. Using a vocabulary of 133 pre-defined semantic concepts, the 111 participants annotated a corpus of approximately 63 hours of video. This corpus was temporally segmented into 46,305 shots and the annotation was performed on a per-shot basis such that the annotators could review one key frame per shot but also replay the full video shot. In addition to the 133 pre-defined semantic concept labels, annotators could assign free-text labels as they saw fit. This led to a total of 1,038 different terms that were used to annotate the complete corpus. The 2003 annotation effort has highlighted several issues regarding manual annotation of large corpora. The annotation that was created was rather sparse: for most of these concept terms too few examples existed to train a semantic model. Lin et al. [2003] report that “107 concepts have more than 100 examples. Only 185 concepts have at least 10 examples.” The VideoAnnEx [Lin et al., 2002] video annotation system that was used is a feature-rich application that forces users to annotate with all available concepts from the lexicon simultaneously. We believe this may have promoted an incomplete annotation of the corpus. Moreover, when we examined the list of annotated labels published by Lin et al. [2003], we observed many redundant labels, such as Dance and Dancing, or Hand and Human Hand.

A second important issue that has not yet been addressed is the ambiguity of visual information. Human reviewers tend to interpret pictures differently and the judgement whether an image or a video shot is a positive or a negative example may differ between individuals. Each video shot that has been annotated in the 2003 annotation effort was reviewed by only a single annotator. This means that a model that is created on the basis of this annotation reflects the individual opinion of only one person. Shatford [1986] explores a theoretical approach to describing pictures for cataloguing purposes and concludes that the interpretation of pictures cannot be consistently indexed. The information in an image is often generic and specific at the same time, and the same image may have different meanings to different people [Shatford, 1986].

We have been involved in conducting a second annotation forum in 2005 which aimed at addressing some of the issues related to ambiguity of visual content. We describe this effort in detail in Chapters 4 and 5. In particular, we address the question how to use multiple non-uniform annotations to model a more generic view of visual content in Chapter 4. In
Chapter 5, we discuss the influence of the concept vocabulary and the mode of annotation on annotation quality and present strategies to maximise annotation efficiency and quality.

2.4 Content-based Video Search

Due to advancements in automatic speech recognition technology [Garofolo et al., 1999], content-based video search can benefit from using the spoken text in the audio track. Semantics in video can be captured by automatic transcription of the spoken information into text [Hauptmann and Smith, 1995], and well-understood techniques from text-document retrieval can then be applied to search the text-transcripts [Hauptmann and Smith, 1995; Smeaton et al., 2002].

2.4.1 Video Search using Spoken Text

Speech-based video search techniques have perhaps provided the most significant benefits for content-based video retrieval [Hauptmann, 2005; Hauptmann and Christel, 2004] as it provides a scalable method for semantic search. A common approach is leveraging the textual information that can be obtained from Closed Captions (CC), Automatic Speech Recognition (ASR), and Optical Character Recognition (OCR) sources [Hauptmann, 2005; Nock et al., 2003a; Over et al., 2005]. Closed captions are frequently unavailable, and video OCR is limited as it applies only to video segments that contain inscriptions in the video imagery. Most videos do carry spoken information. Complemented by the fact that automatic speech recognition is a well understood technique, speech-based retrieval is perhaps the most popular technique used for video search and retrieval [Hauptmann, 2005; Smeaton, 2005].

After temporally segmenting the video stream into the desired unit of retrieval — usually a shot — automatic speech recognition systems are applied to convert the spoken information of the audio track into text [Nock et al., 2003a; Pye et al., 1998]. This text can then be time-aligned to the video segments so that each segment is assigned words that are spoken within the temporal window of the segment [Adams et al., 2002; Amir et al., 2005]. As a result, the video clip is represented as a collection of shot-aligned text documents, and can be searched with existing text-search engine methods [Adams et al., 2002]. This process has been well studied, for example, within the TREC Spoken Document Retrieval (SDR) track [Garofolo et al., 1999], and in the TREC Video Evaluation (TRECVID) [Hauptmann and Christel, 2004; Over et al., 2005; Smeaton, 2005; Smeaton et al., 2003]. SDR focuses on retrieving documents from broadcast news audio recordings that do not contain a visual track. The
process is technically not different from text-based video retrieval but the information need underlying visual information retrieval constitutes a significant difference. In contrast to SDR, a document is considered relevant in video retrieval only if the information need that was specified in the query is visually present in the visual content of the document [Smeaton et al., 2003]. This poses a unique challenge to speech-based video retrieval. For example, when users search for video segments showing aircraft, they might use “aircraft” as a query term. However, the likelihood that shots depicting aircraft actually contain this term as spoken text is small. We often observe a mismatch between the information contained in the spoken text and the information of the visual content because the spoken text does not reflect the visual content well [Hauptmann et al., 2003; Xiangyang et al., 2005]. This effect occurs frequently because the spoken track rarely mentions the background scene of a video.

In addition, the automatic speech recognition systems that are available to extract the spoken text have a considerable word error rate [Witbrock and Hauptmann, 1998]. The word error rate is the sum of inserted, deleted, and falsely substituted words, expressed as a percentage relative to the correct transcript [Chen et al., 1998]. In SDR, the word error rate is reported to have only marginal impact on the retrieval performance as long as it is below approximately 35% [Hauptmann, 2005; Nock et al., 2003b], but word error rates this low are rarely achieved in realistic scenarios. While modern speech recognition systems may achieve word error rates as low as 15% under ideal conditions [Hauptmann, 2005], the word error rate may be as high as 70% to 85% under difficult conditions [Witbrock and Hauptmann, 1998]. Nock et al. [2003b], for example, report that the retrieval performance in their experiments dropped when the word error rate of the transcripts increased.

A disadvantage that the speech-based video search approach shares with text document retrieval is the language barrier [Hauptmann, 1995]. Speech-based search techniques are only designed to work within one language; the query is formulated in the same language as the document collection. Recent research in video retrieval brings attention to this problem, but so far without major success [Kraaij et al., 2006; Over et al., 2005]. The current approach of applying machine translation techniques after native language speech recognition is problematic because the errors of the machine translation are added to the errors of the automatic speech recognition [Hauptmann et al., 2005].

Another effect is that the appearance of events or objects of interest might not be exactly synchronised with their appearance in the spoken text [Iyengar et al., 2002]. A frequent observation in produced footage such as television news, documentary, and other broadcasts is that visual content often only appears after it is referred to in the spoken information [Nock
2.4. CONTENT-BASED VIDEO SEARCH

et al., 2003b]. For example, in news broadcasts, an on-site report is usually introduced by a commentator in the studio, called the \textit{anchor person} [Smoliar and Zhang, 1994]. This temporal misalignment can be addressed by considering spoken text that coincides with the current shot and the immediately surrounding shots when time-aligning the speech text with the segmented video [Adams et al., 2002; Iyengar et al., 2002; Nock et al., 2003b]. As a result, each shot is not only assigned the spoken text that lies within the temporal window of the shot, but also text that is spoken just before and after. Different methods for such text-to-shot alignment have been investigated [Brown et al., 1995; Nock et al., 2003b] and optimal schemes are dependent on the average shot length [Nock et al., 2003b]. However, it is more difficult to alleviate the mismatch in semantics between the spoken and the visual tracks. Speech-based video retrieval thus tends to perform well in answering specific queries about named people, sites, or objects, but it usually fails at generic queries involving unnamed people, objects, settings, or events [Haubold et al., 2006; Hauptmann et al., 2005].

2.4.2 Query Expansion for Speech-based Video Search

Query expansion is a promising approach for addressing problems such as poor recall due to speech terms not matching or misalignment of the spoken information to the visual information [Hauptmann, 2005]. The principle of query expansion is to automatically add terms to the original query that are found to be related to the query. The terms to be added may include synonyms of the original query terms, or non-synonym terms that frequently co-occur with the query terms in the same context, and are therefore topically related, for example, “aircraft” and “airline”. Synonym or hypernym-based query expansion approaches are referred to as \textit{global query expansion} [Xu and Croft, 1996] as they are based on the lexical properties of the language and are corpus-independent. They are frequently based on dictionaries or sources such as WordNet\footnote{http://wordnet.princeton.edu} [Miller, 1995; Voorhees, 1994]. Haubold et al. [2006] have explored lexical query expansion to map query terms to terms from a lexicon of visual concepts. For each visual concept they can then trigger a retrieval step using visual low-level features. The results obtained with this are used to re-rank the speech-based search in a fusion step to improve precision. They report substantial improvements in news video search experiments with this re-ranking approach. Other researchers use similar approaches to expand queries with lexically related terms, for example Chang et al. [2005] and Xiangyang et al. [2005], or to classify queries semantically [Hauptmann et al., 2004].
Conversely, co-occurrence based approaches, are considered local [Attar and Fraenkel, 1977], as they rely on corpus-dependent term co-occurrence and frequency statistics. The typical strategy is the one that has been proposed by Rocchio [1976], which expands the query with terms taken from documents that are considered relevant to the original query. In addition, each query term is assigned a weight factor such that the original terms can be weighted differently to the added terms. This strategy is commonly referred to as pseudo-relevance feedback because the query expansion is performed in a fully-automatic process. The original query is executed and the $N$ top-ranked documents from the result set of this query are analysed to select additional query terms [Xu and Croft, 1996]. The additional query terms are then added to the original query to yield the expanded query; the expanded query is then re-executed to generate the final result set. The term pseudo-relevance feedback is attributed to the fact that the $N$ top-ranked documents are only assumed to be relevant, but any knowledge whether these documents are indeed relevant does not yet exist. Hence, these documents are referred to as pseudo-relevant. Pseudo-relevance feedback may improve recall — especially for short queries — by allowing document matches to additional terms related to the original query. For example, “aircraft” can be expanded with “airline” or “pilot”. It may also narrow down queries that are too broad, such as expanding “car” to “car accident”, thereby re-ranking results and improving precision as long as the refined query is indeed relevant to the original one. Another advantage of this method is that it helps to discover relevant terms that do not necessarily have a lexical relationship with the original query terms, but frequently co-occur with them. Yan et al. [2003] report promising results with pseudo-relevance feedback in speech-based video retrieval experiments using relatively noisy text-transcripts from automatic speech recognition. However, experiments in text-document retrieval have shown that query expansion is highly query-dependent [Xu and Croft, 1996] and bears the risk of topic drift [Carmel et al., 2002; Hawking, 2000]. Topic drift refers to the effect that search results may become more and more irrelevant to the original query when the pseudo-relevance assumption does not hold true during the query expansion process.

Carmel et al. [2002] propose query expansion using Lexical Affinities (LA) to minimise the problem of topic drift. While this approach is also based on pseudo-relevance feedback, it employs an alternative term selection method that aims to improve precision by selecting terms that increase the specificity of the query. They consider lexical affinities, which are

---

17This method is often referred to as “Rocchio query refinement”.

45
pairs of terms that frequently co-occur within a close proximity of each other. This may be within one phrase or within a few words in a sentence. Word pairs such as “car accident” or “aircraft pilot” would be considered lexical affinities. If one part of a lexical affinity is found in the query text, it is assumed that the other part of the LA is also relevant. For example, the term “car” in the original query would then be expanded to “car accident”. Carmel et al. [2002] report improvements in precision of up to 15% in fully-automatic text document search experiments [Carmel et al., 2002]. A disadvantage that all pseudo-relevance feedback methods share is that they require several collection- and the query-dependent parameters [Carmel et al., 2002; Xu and Croft, 1996] to be set. These are the number of top-ranked documents to consider, the maximum number of terms to add to the original query, and the weight of the added terms in relation to the original query terms.

Another method to refine textual queries attempts to prevent topic drift by disambiguating word senses using semantic text annotation. In this approach, text is analysed using Part-of-speech tagging [Klein and Simmons; Merialdo, 1994] and named-entity detection. Part-of-speech tagging is the process of identifying grammatical word forms in sentences, such as nouns, verbs, and adjectives. Named-entity detection can then be used to identify known objects, places, or people using a dictionary. These techniques are often used for question answering tasks [Carroll et al., 2004; Wu et al., 2002] or for query classification in video retrieval systems [Hauptmann et al., 2004; Wu et al., 2002; Yan et al., 2004]. At indexing time, the entire corpus can thus be analysed to automatically detect and annotate terms with semantic categories, such as people, roles, objects, places, events, and program categories [Prager et al., 2000]. The query can be analysed in the same way at query time to be disambiguated and reformulated. Document relevance during the retrieval process can then be estimated not only on the basis of terms matching, but also on the basis of matches of the semantic annotation. For example, a query containing the term “basketball” may automatically be annotated with the Sports category, “car” may be annotated with Vehicle, while “George Bush” can be annotated as Person and President. This approach has the potential to allow semantic refinement of query topics, while limiting topic drift. However, it is limited to the set of semantic categories that can be annotated reliably.

In Chapter 6, we explore several query expansion and refinement techniques when applied to video retrieval. Moreover, we present a fusion approach that combines concept terms that were automatically detected from the visual content with a lexical expansion technique to arrive at a multimodal retrieval system allowing free-text queries.
Figure 2.8: After shot segmentation one key frame and usually several low level features are extracted for each shot. As part of the indexing step, semantic concepts are detected on the basis of the low-level features. The search index allows the user to search using free-text queries, sometimes in combination with example images as part of the query.

2.4.3 Multimodal Search

Video retrieval benefits much from speech-based search techniques, but problems and limitations remain. Only multimodal search approaches can utilise the comprehensive information that video contains [Baan et al., 2001; Mezaris et al., 2004] in the form of sound, speech, and visual content. The current state-of-the-art approach to multimodal video search is schematised in Figure 2.8. First, shot segmentation and key frame selection are applied. Low-level features of each shot are then extracted, and automatic concept detection is employed to map semantic concepts to the low-level features [Campbell et al., 2006; Chang et al., 2006]. Finally, an index is generated that allows querying by utilising speech transcripts and the semantic concept terms [Snoek et al., 2006a]. This may be combined with visual search that allows users to select images as query examples [Campbell et al., 2006]. In the visual search technique, low-level features of the query examples and the key frames are compared directly.

Effectively combining the information from the different modalities during index generation is a difficult task, in particular in a fully-automatic search approach [Hauptmann and Christel, 2004]. In the context of English language video retrieval, Smeaton [2005] summarises: “text search [...] continues to be the single most important modality for video,
2.4. CONTENT-BASED VIDEO SEARCH

being far more important than retrieval based on visual features”. Hauptmann and Christel [2004], and Smeaton [2005] report that multimodal video search approaches work well utilising interactive strategies with relevance feedback that allows users to iteratively refine their search. However, according to these authors, video retrieval without any user interaction is not yet useful for practical application.

Recently, generalised multimodal search approaches have been presented that achieved promising results [Campbell et al., 2006; Cao et al., 2006; Chang et al., 2006; Snoek et al., 2005; 2006a]. These systems allow free-text or concept-based querying of video databases and leverage visual, text, and audio features through automatic concept detection using machine learning techniques. While the results that these systems achieve are promising, they also highlight issues that still require further research. For example, automatic concept detection helps as long as many different concepts can be trained with sufficient detection accuracy, and as long as enough training examples for each exist [Snoek et al., 2006a]. Automatic concept detection alone does not guarantee good results and the fusion of the information from different modalities is both important and difficult [Chang et al., 2006; Snoek et al., 2006a]. Interactive search approaches still far outperform fully-automatic systems [Smeaton and Ianeva, 2006], and the problem of generic and automatic video retrieval is still not solved.

2.4.4 Performance Evaluation

To our knowledge, the TREC Video Retrieval Evaluation (TRECVID) that we described previously is the only large-scale performance evaluation for video retrieval systems that provides a standardised test bed. Consequently, we use the TRECVID test and development corpora for our evaluation and adopt the performance evaluation scheme that is used at TRECVID [Smeaton et al., 2001; Voorhees and Harman, 2001]. Detailed Information on the search test corpora that we have used is provided in Appendix C; these include the test sets of the years 2003, 2005, and 2006. The evaluation measure used for the search task at TRECVID is average precision [van Rijsbergen, 1979], a common measure in information retrieval. NIST provides the trec_eval program\textsuperscript{18} to calculate mean average precision and other statistics for evaluation. Average precision combines recall and precision into one value by averaging the precision over all relevant returned results. Let $\rho^k = \{i_1, i_2, \ldots, i_k\}$ be a set of results in which each $i$ represents one shot, ranked by their estimated probability to be relevant. We refer to this set as the answer set $A$. We further define for any rank $k$, $R \cap \rho^k$

\textsuperscript{18}http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval_video
to be the number of relevant shots in the top \( k \) answers of \( \rho \), where \( R \) is the total number of relevant shots in the collection. The average precision is then defined as:

\[
P = \frac{1}{R} \sum_{k=1}^{A} \frac{R \cap \rho^k}{k} \text{rel}(i_k)
\]  

(2.18)

where the relevance indicator function \( \text{rel}(i_k) = 1 \) if \( i_k \in R \) and 0 otherwise. This measure favours higher-ranked relevant shots over lower-ranked relevant shots because the denominator \( k \) and the value of the relevance indicator function are dominant. We believe that this is sensible as it is more useful to a user if relevant results appear highly ranked. Up to 1000 shots per query are evaluated in the search task at TRECVID. To evaluate retrieval performance across multiple queries, TRECVID uses Mean Average Precision (MAP) [van Rijsbergen, 1979], which is the mean over the average precision of multiple queries.

Each year, NIST provides a development collection of video data that includes common shot boundary definition and speech-transcripts from automatic speech recognition and closed captions (where available). This ensures that participants have a common basis to develop and test their approaches. The actual test corpus including a set of query topics is only released shortly before result submission is due. All participants are then asked to perform blind runs on the test corpus and submit between six and ten\(^{19} \) result sets for each query topic. Example query topics may be found in Table 6.1 on page 135 and in Table 6.4 on page 151. Until 2003, two search strategies were defined in which groups could participate: interactive and manual. In interactive search, each user is allowed up to fifteen minutes on an interactive search system to produce a result set; the user may specify and refine the query as they see fit. In manual search, each user is allowed to formulate a query based on the given topic and run this query to produce the result set without further interaction. Since 2004, NIST has added the fully-automatic search strategy in which a search system has to process the query topics as given by NIST without any user interaction.

Since 2003, NIST categorises all participating search approaches based on the training data that have been used during development phase. This has mainly been introduced because of the common high-level feature annotation that has been made available from the annotation forums. To keep results comparable, participants have to specify which data they have used when developing and training their system. NIST defines the following three categories [Smeaton et al., 2003]:

\(^{19}\)The allowed maximum differed for each year depending on the number of participants.
2.5. SUMMARY

**Type A:** Systems that only use common TRECVID development collection data, such as the past TRECVID corpora.

**Type B:** Systems that use common development collections but include additional data such as semantic annotations that go beyond those that were generated during the annotation forum.

**Type C:** Any Systems that are not of type A or type B.

2.5 Summary

In this chapter, we discussed elementary aspects of representing and accessing content as digital video. Different tasks require different representations in terms of colour, spatial and temporal layout. To identify shot boundaries in video, we need to access each individual frame. While shot boundary detection does not require deep understanding of the semantics in the video, we must be able to compare individual frames and distinguish scene activity from shot transitions. This is particularly difficult for gradual transitions as these involve subtle inter-frame changes over several frames and the seemingly simple problem of identifying transition effects reliably remains a challenging task.

In contrast, semantic classification of video shots is more concerned with the meaning that video content conveys to humans. As machines cannot actually understand this meaning, we classify unseen video segments by comparing these to pre-classified examples on the basis of low-level features. The process of classifying the examples is an important aspect because this has an influence on how well we can classify unseen data. Moreover, for audiovisual content such as video, clear semantic classification is a difficult task, even for humans. To provide generic approaches to the classification problem, we should model human interpretation of audiovisual content over a large population sample. While it has not yet been identified how this modelling can be done, obtaining multiple opinions for a large collection of videos is also a challenge that requires careful planning. The goal is to maximise the outcome in terms of quality and quantity, while utilising human labour optimally.

Shot boundary detection and semantic classification facilitate content-based video search using concept terms. This is complemented by automatic speech recognition which is used to transcribe spoken information into text. While one question is how to combine semantic terms and spoken text in a search approach, another question is how to address some of the shortcomings from which text-based video search suffers. Reformulating query text and
expanding words and phrases with related terms may be used to alleviate some of these shortcomings, but this may introduce new problems such as topic drift and topic dependency. Text-based search on written documents is a well-understood technique, but applied to video, it poses new challenges that need to be addressed.
Chapter 3

Temporal Video Shot Segmentation

“There is no problem so complicated that you can’t find a very simple answer to it if you look at it right. Or put it another way: The future of computer power is pure simplicity.”
– The Salmon of Doubt by Douglas Noël Adams

Enabling access to specific information in video streams, video data mining, and retrieval requires the decomposition of video clips into smaller entities. A common first step is therefore to segment a video into shots, where a shot is defined as a contiguously recorded sequence of frames [Konigsberg, 1989]. During the editing stages of a video, shots are concatenated using transition effects. Shot segmentation, or shot boundary detection, can be seen as reversing the video editing step; it is an integral part of the layout reconstruction step [Snoek and Worring, 2005], and fundamental to multimodal video analysis.

Video shots as a basic unit carry usually some coherent semantic information. Several shots may be combined into stories which are continuous blocks of storytelling concerning a specific topic, location, or character. Shots are therefore a common unit for video storage and retrieval; the task of segmenting videos into shots is crucial to effective handling of digital video content. Because of its fundamental impact on advanced video processing, it is important to maximise the effectiveness and efficiency of shot segmentation algorithms. As discussed in Chapter 2, shot segmentation is a well-studied area of research. While many segmentation algorithms have been proposed, they have mostly not been evaluated with large test collections modelling a realistic scenario. Modern television footage that contains a variety of visual effects and can be described as “fast cut” poses new challenges to transition detection algorithms. Despite this area being well-studied, the video shot segmentation problem cannot be regarded as solved [Smeaton et al., 2003].
In this chapter, we present our algorithm for shot boundary detection that examines several consecutive frames within a moving window. This algorithm uses the frame ranking approach by Tahaghoghi et al. [2002] for cut detection, which we improve upon by using a localised histogram feature and dynamic thresholds to enhance scalability and effectiveness. Our main focus, however, is on detecting gradual transitions. While these are increasingly popular in modern production footage, gradual transitions are more difficult to identify than cuts. We propose a novel algorithm using the average frame similarity over several consecutive frames for effective detection of gradual transitions. The remainder of this chapter is organised as follows: we describe the application of localised histograms and dynamic threshold computation to improve the effectiveness of cut detection in Section 3.1. In Section 3.2, we describe our novel approach for effective gradual transition detection that uses average frame similarity. In Section 3.4, we discuss experimental results that we obtained when we applied these algorithms to the TRECVID shot boundary detection test sets. We conclude the chapter with a summary of our findings in Section 3.5.

3.1 Improvements on Histogram-based Cut Detection

As described in Section 2.2.2, Tahaghoghi et al. [2002] proposed the ranking technique in a moving window of frames for video shot segmentation, and show this to be very effective for cut detection with a feature derived from the wavelet transform of the frame data. However, they observe this feature to be relatively expensive to compute, and that it produced poor results in gradual transition detection.

Colour histograms, on the other hand, are widely used for shot boundary detection [Koprinska and Carrato, 2001; Smeaton and Over, 2002] and are relatively simple to compute. As the results by Tahaghoghi et al. [2002] show, the best combined performance for cut and gradual transition detection was achieved with HSV colour histograms. While we aim at developing an effective gradual transition detection technique, we wish to maintain effective cut detection to arrive at a combined approach for shot boundary detection. Ideally, we achieve this by using only a single low-level feature to keep the algorithm complexity low. We thus focus on using HSV histograms.

3.1.1 Low-level Feature and Distance Metric Selection

Tahaghoghi et al. [2002] used global HSV histograms in their experiments; one histogram represents each frame. As discussed in Section 2.2, several researchers have proposed to divide
3.1. IMPROVEMENTS ON HISTOGRAM-BASED CUT DETECTION

Figure 3.1: We generate localised histograms by dividing each frame into sixteen equal-sized regions and extracting a one-dimensional HSV histogram for each region. We then concatenate the sixteen histograms into one feature vector for further processing.

frames into regions and to extract a histogram for each region. These localised histograms have been reported to yield improvements in shot boundary detection [Boreczky and Rowe, 1996; Nagasaka and Tanaka, 1992]. On the other hand, extracting several three-dimensional histograms such as those described in Section 2.1.3 per frame can lead to large feature vectors. For example, using 32 bins for each of the three colour components with sixteen frame regions leads to a 524,288-dimensional feature vector per frame (32 · 32 · 32 · 16 = 524,288). Such high-dimensional feature vectors may not be useful due to their sparsity. That is, many histogram bins may be empty or near zero.

To reduce the feature vector size — and minimise the likelihood of generating sparse histograms — we use a one-dimensional histogram representation, as illustrated in Figure 3.1. We divide each frame into 4 × 4 equal-sized regions and extract one 32 bin histogram per colour component for each region. We then concatenate the three histograms for each colour component to form a 96-dimensional vector (32 + 32 + 32 = 96) representing each frame region. This results in a 1,536-dimensional feature vector per frame when using sixteen frame regions (96 · 16 = 1,536). In preliminary experiments, we determined that using 32 bins per colour component represents a good trade-off between computational complexity and detection performance. Smaller bin sizes, that is more bins per colour component did not yield significant improvements in detection performance. However, we do not report on these experiments here.
Ma and Zhang [1998] report good shot boundary detection results when using localised HSV histograms in combination with the $L_1$ norm (Manhattan) distance metric. We could confirm this in our own preliminary experiments in which we compared the $\chi^2$-test comparison [Nagasaka and Tanaka, 1992] and the $L_2$ norm with the $L_1$ norm. While we do not report details of these experiments here, we achieved best results using the $L_1$ norm distance measure to compute histogram differences. We have thus used this measure in all our subsequent experiments. From the generalised Minkowski norm $L_M$, as defined in Equation 2.9, we define the $L_1$ norm for the two feature vectors $\vec{a}$ and $\vec{b}$ of dimension $K$ as:

$$L_1(\vec{a}, \vec{b}) = \sum_{k=0}^{K-1} |a_k - b_k| \quad (3.1)$$

We use this distance metric over the full feature vectors; each region is included with an equal weight; we omit an averaging step because we only compare frames with an identical number of regions. We also tested other frame comparison techniques. For example, we ignored the eight largest differences between corresponding histogram regions, as proposed by Nagasaka and Tanaka [1992]. But we could not achieve any improvements using this technique in our preliminary experiments. We also experimented with ignoring the regions in the centre of frames by using a weighting scheme when computing frame differences. While we could achieve improvements on the TRECVID 2004 collection, this scheme has not shown to yield improvements on any other of our test collections.\footnote{Our most recent experiments show that improvements that we reported previously [Volkmer et al., 2004b] were mostly due to using localised histograms and not due to ignoring the frame centre.} Indeed, for some collections, this approach may even lead to reduced performance, and so we use the unweighted frame distance across all regions.

### 3.1.2 Dynamic Threshold Computation

In addition to the relative thresholds used in their ranking approach, Tahaghoghi et al. [2002] applied a global threshold to the distance between the last frame of the previous shot and the first frame of the new shot if a possible cut is detected. This threshold was set to 25\% of the maximum possible inter-frame distance [Tahaghoghi et al., 2002]. We experimented with different settings for this threshold in our experiments, and found it to be dependent on the collection, on the setting of other algorithm parameters, on the histogram feature that is used, and on the distance measure. For example, on the TRECVID 2006 collection, we used values ranging from 8\% to 42\%, achieving good results in combination with different...
settings for other algorithm parameters. This shows that an optimal fixed value for this threshold is difficult to determine a priori, and that a fixed threshold prevents the algorithm from automatically adapting to different types of video.

To alleviate this deficiency, we propose computing a dynamic threshold that is based on the average inter-frame distance of the previously examined frames. While advancing through a video clip, we maintain the average inter-frame difference of all frames that are passed, omitting frames that border cuts or that are part of a gradual transition. We thus compute an average of the inter-frame distance that does not include the usually higher inter-frame differences that accompany shot transitions. In our implementation of the cut detection algorithm, we require that the inter-frame difference of a possible cut be higher than our global average inter-frame difference, replacing the fixed threshold used by Tahaghoghi et al. [2002]. As we discuss below, we use the same threshold scheme as a second criterion in our gradual transition algorithm, comparing the frames that border a possible gradual transition.

In Section 3.4, we show that the ranking approach can be highly effective for cut detection using the localised histogram feature and our dynamic threshold. However, our primary contribution is a novel algorithm for effective gradual transition detection that we now describe.

3.2 Gradual Transition Detection using Average Frame Similarity

The method of ranking frames in a window of several frames works well for abrupt transitions because these usually show significant inter-frame distances within a few consecutive frames. Our observations have shown that this is not usually the case for gradual transitions, where inter-frame distances are typically smaller. This results in the ranking approach being far less effective in detecting gradual transitions.

To address this problem, we propose that the frames in the moving window not be examined individually. Figure 3.2 illustrates a moving window of frames as it traverses a dissolve transition. We define two sets of frames, the pre-frames and the post-frames, that are directly preceding and following the current frame \( f_c \). The pre-frames and post-frames form the moving window of size \( Q \); to ensure an equal size of the sets on either side of the current frame, we specify the window size \( Q \) such that \( Q = 2W \), where \( W \) is the half-window size.

In the example in Figure 3.2, the half-window size is \( W = 5 \), resulting in a window size of \( Q = 10 \) because the current frame is not considered part of the window. For each of the two
sets, we determine the histogram distance between each frame in that set and the current frame while the window advances through the video frame-by-frame. Instead of ranking individual frames, however, we average the intra-set distances, giving a final value that is the average distance between that set and the current frame. This computation results in two values, one each for the pre- and post-frame sets, and we use the ratio of these values — referred to as Pre-Post Ratio $R_{pp}$ — to detect gradual transitions. Handling the frames surrounding the current frame this way caters for the inter-frame differences during gradual transitions accumulating over several frames. Let $\vec{h}_c$ be the histogram feature vector of the current frame $f_c$, the Pre-Post Ratio $R_{pp}$ can then be computed as follows:

$$R_{pp} = \frac{\sum_{i=1}^{W} L_1(\vec{h}_c, \vec{h}_{c+i})}{\sum_{i=1}^{W} L_1(\vec{h}_c, \vec{h}_{c-i})} \quad (3.2)$$

where $L_1(\vec{h}_c, \vec{h}_{c+i})$ is the $L_1$ norm distance between the histogram feature vectors of the current frame $f_c$ and the frame $f_{c+i}$ within the moving window, as given by Equation 3.1.

We use an example section of a video from our test collection to explain our approach in detail: Figure 3.3 shows a dissolve between the neighbouring shots A and B. For this example, we choose the half-window size to be $W = 8$, indicated by blue borders in Figure 3.3. We assume that the dissolve starts at frame 8 and ends with frame 15. In the top row, frame 7 is the current frame, indicated by a yellow border. The current frame belongs to shot A and is the last frame before the transition starts. Frames 1 to 6 are part of the pre-frames together with the two previous frames that are not shown in this figure; the pre-frames are also from shot A. These are similar to frame 7, and therefore, their inter-frame distance to the current frame
3.2. GRADUAL TRANSITION DETECTION USING AVERAGE FRAME SIMILARITY

Figure 3.3: An example of a dissolve transition. The yellow border indicates the current frame, while the blue border indicates pre- and post frames of the moving window. We chose a half-window size of $W = 8$ for this example. Before the current frame enters the transition, $R_{pp}$ is minimal. It rises to a maximum as the moving frame window proceeds through the transition and falls again after the transition. The maximum indicates the end of the transition.
frame is relatively low. For this example, let us assume the average inter-frame distance of the pre-frames to the current frame has the value 2.

Frames 8 to 15 — the post-frames in the first row in Figure 3.3 — are mostly dissolve frames, and therefore relatively dissimilar to the current frame. Hence, the average inter-frame distance for the post-frames is comparatively high; let us assume it has the value 10. Given a pre-frame average of 2 and a post-frame average of 10, the pre-post ratio of the top row in Figure 3.3 is \( R_{pp} = \frac{2}{10} = 0.2 \).

As the current frame moves further into the dissolve, the ratio rises. This is illustrated in rows two and three of Figure 3.3. In the fourth row, the current frame is the last frame of the transition. This frame is likely to be very similar to the following frames that belong to shot B, producing a low average inter-frame distance. For our example, let us take this value to be 2. The pre-frame window formed by frames 7 to 15 are mostly the frames of the dissolve. As we established earlier, their average inter-frame distance is high, we again assume a value of 10. We can now calculate the pre-post ratio for row four as \( R_{pp} = \frac{10}{2} = 5 \). Once the window exits the transition completely, the ratio usually reverts to a relatively low value.

We observed that this behaviour is common for dissolves and fades. By monitoring \( R_{pp} \) as we advance through a video clip, we can detect the minima and maxima that accompany the start and end of such transitions. Other effects, such as wipes and page translations, are more complex, and often include intense motion. Such transitions can also be detected using our approach, but with reduced effectiveness.

We maintain a history of \( R_{pp} \) values to calculate a moving average for peak detection. Based on this moving average and the standard deviation, we compute a threshold that we multiply by a factor that we call the Threshold Factor \( T \).

Figure 3.4 shows the \( R_{pp} \) curve for a 200-frame segment of a video, along with the corresponding moving average and threshold. A possible gradual transition is indicated if \( R_{pp} \) crosses the threshold. In this case, we determine the positions of the local maximum after the threshold was crossed, and the position of the local minimum within the preceding frames. A gradual transition is reported over the interval between these two points. However, as for cut detection, a gradual transition is only reported if the last frame of the previous shot and the first frame of the next shot are sufficiently different. For this second criterion, we use the same definition as for cut detection and require these two frames to have a greater difference than the average inter-frame difference of the previous frames.
3.3 Algorithm Parameters

We integrate cut detection using the ranking approach of Tahaghoghi et al. [2002] and gradual transition detection with our method. This has the benefit that both can be handled in a single pass through a video clip using one moving window of frames. The primary parameter of our algorithm is the size of the moving window $Q$ that is specified by the half-window size $W$. As discussed above, the current frame $f_c$ is not considered part of the window, and so the window size is $Q = 2W$. Our experiments show that the optimal settings for $W$ for gradual transition detection depend on the average length of the transitions in the collection. We could not observe any such collection-dependency for cut detection. Moreover, effective cut detection requires only a small half-window size.

To cater for the different optimal window sizes, our algorithm uses the full window for gradual transition detection and only a part of the window for cut detection. We therefore specify $W_g$ to define the full window size and, in addition, define $W_c$ such that $W_c \leq W_g$.

The upper bound $U_c$ and lower bound $L_c$ thresholds that the cut detection requires are set to $U_c = W_c - 1$ and $L_c = 2$, where $L_c < U_c$. This implies that the half-window size for cut detection can reach a minimum value of $W_c = 4$. An advantage of the ranking approach is that these parameters do not need any further adjustment because they are largely independent of the video footage that is being processed. In all our experiments, we
achieved the optimal results when setting the upper bound and lower bound as described above. These parameters may therefore be regarded as part of the algorithm specification.

For gradual transition detection, our algorithm requires the threshold factor $T$ to be set for computing the threshold for peak detection. This factor determines how much above the moving average the threshold is set. Common values range from 1.5 to 2.5 and depend on the video footage. Videos with much activity and rapid motion tend to cause a noisier $R_{pp}$ curve and usually require a value for $T$ that is close to or above 2.0 for optimal results.

A second parameter that may be varied to influence the effectiveness of the peak detection is the threshold history size factor $M$. The history size factor $M$ is multiplied by the window size $Q$ to determine the final size of the history buffer to compute the moving average for $R_{pp}$. The default is $M = 1$ so that the buffer size normally equals the size of the window $Q$. $M$ may be manually altered so that $M \in \mathbb{N}$, with $1 \leq M \leq M_{\text{max}}$, where $M_{\text{max}}$ is the maximum possible value implied by the length of the video clip. A larger history size buffer causes a smoother moving average which may be useful to adjust for a noisy $R_{pp}$ curve. More details on the choice of these parameters are discussed in the next section.

### 3.4 Performance Evaluation

We evaluate our system with the TRECVID shot boundary detection test sets from the years 2001 to 2006. Each of these test collections comprises up to 22 video clips of educational, promotional, documentary, and television news footage. The total duration of each of the test sets ranges between approximately five and eight hours, featuring up to 4,800 shot boundaries each. Detailed information on these test sets can be found in Appendix A.

#### 3.4.1 Methodology

We measure transition detection performance by evaluating recall $R$, precision $P$ [Ruiloba et al., 1999], and the quality measure $Q$ [Quénot and Mulhem, 1999]. To evaluate the performance in correctly detecting start and end of gradual transitions, we use frame-based recall $R_F$ and frame-based precision $P_F$. These measures are described in Section 2.2.3.

To make our results comparable with those reported at the annual TRECVID workshops, we follow the evaluation methodology that is used at TRECVID. As described in Section 2.2.3, the reference specifies the four transition types, $\text{Cut (CUT)}$, $\text{Dissolve (DIS)}$, $\text{Fade-out/-in (FOI)}$, and $\text{Other (OTH)}$. However, the evaluation only distinguishes between gradual transitions and abrupt transitions. This means that reported cuts can match only
3.4. PERFORMANCE EVALUATION

cuts in the reference, but all other transitions may be reported as *Gradual* (GRA) and can be matched interchangeably against any transition that is not a cut. Gradual transitions that are shorter than six frames — also called *short graduals* — are an exception. Short graduals are perceived as cuts by humans and the evaluation allows for them to be matched against both detected gradual transitions and detected cuts.

3.4.2 Experiments and Discussion

To identify improvements that may be achieved by using the localised histograms instead of the global histograms, we include results on both feature representations in this section. We use the HSV colour space in all our experiments and thus refer to the global histogram feature as *HSV histogram*, and to the local histogram feature as *L-HSV histogram*. Both histograms used a one-dimensional representation as described in Section 3.1.1. We did not focus on parameters of the histogram features such as the bin size or the number of regions in our histograms. Consequently, preliminary experiments that we conducted to determine good choices for histogram bin size and the number of regions for the localised histogram feature are not discussed here as they are not specific to our algorithm. We used 32 bins per colour component in all histograms; for the localised histograms, we used $4 \times 4$ regions per frame.

Table 3.1 shows results that we obtained on the TRECVID test sets using the low-level features HSV and L-HSV. The parameters were set to obtain maximum quality for all transitions. We determined these parameter values empirically. Our system achieves strong results, well above 80% quality for most test sets. The 2002 and 2006 test sets are more difficult for our algorithm and we observe weaker results for these sets. The 2002 test set is largely comprised of older analogue video of low quality. This results in a very noisy $R_{pp}$ curve that is problematic to handle. As can be seen from Table 3.1, the 2002 collection is the only one for which we achieved better results when increasing the threshold history size factor to $M = 3$ to smooth the moving average curve. For all other sets, a threshold history size that equals the size of the moving window is optimal.

Not surprisingly, the optimum half-window size $W_g$ for gradual transition detection correlates with the average length of gradual transitions in the test collection. Due to the hypothesis underlying our gradual transition detection algorithm, our system is most effective if a gradual transition fills approximately one half-window of the moving window. Larger transitions can still be handled well, but transitions that are much shorter than the size of
Table 3.1: Performance of our algorithm measured by Recall $R$, Precision $P$, and Quality $Q$ on the TRECVID shot boundary test sets, along with the algorithm parameters. The parameters were set to maximise the quality index for all transitions. Parameters for gradual transition detection need to be tuned to obtain optimal results, while $W_c$ need not to be varied across different collections. In most cases, we obtain the best results using the localised HSV histogram feature (L-HSV).

one half-window constitute a problem. In this case, the peak in the $R_{pp}$ curve that normally marks the end of a transition occurs only well after the end of the transition. Our algorithm then reports the end of the transition incorrectly, which may cause it to be counted as a false positive in the evaluation. This is frequently the case for short gradual transitions (shortgraduals) that extend over five frames or less, as explained in Section 3.4.1. When comparing the test set statistics that are listed in Appendix A, we observe that the more recent test collections include a substantial number of short graduals. In the 2006 test collection, more than 47% of all gradual transitions are short graduals, which explains the poorer detection quality of our algorithm on this test set.

As the separated results for cuts and gradual transitions in Table 3.1 show, gradual transition detection is a more difficult problem than cut detection, and the results are generally weaker. This difference is particularly marked on the TRECVID 2006 test set because of the large number of shortgraduals in this collection, and to a smaller extent on the 2005 test set in which 35% of all gradual transitions are shorter than six frames. The L-HSV feature is more robust to different types of video between the collections; however, on the 2006 collection, the global HSV histogram feature outperforms the localised histogram feature in cut
3.4. PERFORMANCE EVALUATION

### Feature: HSV

<table>
<thead>
<tr>
<th>Test set</th>
<th>All transitions</th>
<th>Cuts</th>
<th>Gradual transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deleted</td>
<td>Inserted</td>
<td>Deleted</td>
</tr>
<tr>
<td>2001</td>
<td>0.066</td>
<td>0.191</td>
<td>0.047</td>
</tr>
<tr>
<td>2002</td>
<td>0.121</td>
<td>0.345</td>
<td>0.108</td>
</tr>
<tr>
<td>2003</td>
<td>0.129</td>
<td>0.183</td>
<td>0.105</td>
</tr>
<tr>
<td>2004</td>
<td>0.091</td>
<td>0.172</td>
<td>0.070</td>
</tr>
<tr>
<td>2005</td>
<td>0.094</td>
<td>0.240</td>
<td>0.053</td>
</tr>
<tr>
<td>2006</td>
<td>0.170</td>
<td>0.285</td>
<td>0.118</td>
</tr>
</tbody>
</table>

### Feature: L-HSV

<table>
<thead>
<tr>
<th>Test set</th>
<th>All transitions</th>
<th>Cuts</th>
<th>Gradual transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deleted</td>
<td>Inserted</td>
<td>Deleted</td>
</tr>
<tr>
<td>2001</td>
<td>0.059</td>
<td>0.124</td>
<td>0.030</td>
</tr>
<tr>
<td>2002</td>
<td>0.115</td>
<td>0.176</td>
<td>0.067</td>
</tr>
<tr>
<td>2003</td>
<td>0.112</td>
<td>0.143</td>
<td>0.080</td>
</tr>
<tr>
<td>2004</td>
<td>0.100</td>
<td>0.130</td>
<td>0.069</td>
</tr>
<tr>
<td>2005</td>
<td>0.105</td>
<td>0.176</td>
<td>0.054</td>
</tr>
<tr>
<td>2006</td>
<td>0.197</td>
<td>0.276</td>
<td>0.140</td>
</tr>
</tbody>
</table>

Table 3.2: Analysis of deleted and inserted transitions for the runs shown in Table 3.1. Deleted transitions are existing transitions that our algorithm did not detect (false negatives) and inserted transitions are false positive detections. Most difficult are the test sets from 2002 and 2006 for which our algorithm reports many false positive gradual transitions.

detection. We observed that many video clips in this collection include sections converted from 16:9 aspect ratio to 4:3 aspect ratio, with black areas at the top and the bottom of each frame. The L-HSV feature appears to be less effective for these sections. This could be handled by cropping out the monochrome areas of all affected frames and re-scaling them to the 4:3 format.\(^2\) However, we did not apply any pre-processing of the videos and will consider such an approach in our future work.

Table 3.2 shows an analysis of false positive and false negative detections for the runs from Table 3.1. Our system has generally high recall, expressed in the low rate of deleted transitions. The rate of false positives for cuts is low; the rate of false detections for gradual transitions is much higher, especially for the 2002 and the 2006 test sets. Again, the L-HSV feature performs generally better than the global HSV histogram feature. The tendency of our gradual transition detection algorithm to falsely detect too many transitions is the reason for the low precision that we observe with some test sets. However, as Quénot and Mulhem [1999] point out, from an application point of view, false positive detections are less detrimental as they could be filtered out in a subsequent processing step.

\(^2\)In contrast to pan-scanning, this operation would change the aspect ratio but retain the full frame content.
Table 3.3: Detailed analysis of the gradual transition detection performance of our algorithm for the runs shown in Table 3.1. Our system is most effective at detecting dissolves (DIS) and fade-out/-in (FOI) transitions. Other gradual transitions (OTH) are found only with reduced effectiveness. Frame-recall for fade transitions is relatively low compared to the frame-recall results for dissolves and other gradual transitions.

Table 3.3 shows more detailed results for gradual transition detection, separately for the three transition types that are specified in the reference data. The hypothesis for gradual transition detection underlying our algorithm is largely based on the paradigms of dissolve and fade transitions. As a result, our system detects these with better effectiveness than other gradual transitions. Dissolve and fade transitions form the vast majority of all gradual transitions in our test collections, as can be seen from the statistics in Appendix A.

Frame-based recall for fade transitions is on average much lower than frame-based recall for dissolves and other gradual transitions. The reduced frame-recall for fade transitions means that the system reports the length of the transition relatively well, while the locations of the start and the end are not reported as accurately. The detected fades tend to be reported with an offset to the reference. This is due to the average length of fade transitions being longer than the average length of dissolves and other transitions in most test collections. However, fade transitions are relatively rare and account for less than 5% of all transitions in our test collections.

Figure 3.5 illustrates the impact of varying the half-window size $W_g$ on gradual transition detection performance. This graph shows results obtained with the TRECVID 2004 test
3.4. PERFORMANCE EVALUATION

![Graph showing the influence on quality, recall, and precision of varying half-window size \( W_g \) for gradual transition detection with the L-HSV histogram feature on the 2004 test set. Recall and quality decrease with smaller window sizes; we observe optimum quality at \( W_g = 18 \). Precision remains stable and even improves with very small window sizes.](image)

**Figure 3.5:** The influence on quality, recall, and precision of varying half-window size \( W_g \) for gradual transition detection with the L-HSV histogram feature on the 2004 test set. Recall and quality decrease with smaller window sizes; we observe optimum quality at \( W_g = 18 \). Precision remains stable and even improves with very small window sizes.

...collection, using the L-HSV feature. We set the half-window size for cut detection to the optimal value \( W_c = 6 \) that was empirically determined in previous experiments. Parameters that affect gradual transition detection were set to the optimal values \( T = 1.6 \) and \( M = 1 \); we then varied \( W_g \) to plot the graph that is shown in Figure 3.5. The quality index correlates strongly with recall because, as shown in Equation 2.15, it favours recall over precision. Both quality and recall decrease significantly once the half-window size becomes much smaller than the optimum value of \( W_g = 18 \). At the same time, precision remains relatively stable and improves for smaller half-window sizes. Using larger window sizes does not have a significant positive or negative effect on detection performance.

The graph in Figure 3.6 shows the results for varied half-window sizes \( W_c \) for cut detection using the L-HSV feature on the TRECVID 2004 test set. We used the optimal settings for gradual transition detection \( W_g = 18 \), \( T = 1.6 \) and \( M = 1 \) that we empirically determined previously. We then varied \( W_c \) such that \( 4 \leq W_c \leq W_g \). As explained in Section 3.3, our implementation does not allow \( W_c \) to be larger than \( W_g \) and the minimum value of \( W_c = 4 \)...
is given by the requirement of $L_c < U_c$. For the TRECVID 2004 test collection, the smallest possible value $W_c = 4$ produces the best recall and therefore the best quality, but precision decreases for half-window sizes smaller than $W_c = 6$. Across all the collections that we used, $W_c = 6$ generally produces the best quality. Recall and quality decrease significantly for half-window sizes much larger than $W_c = 10$.

We determine the setting of the threshold factor $T$ empirically on the basis of training experiments. Optimal settings may vary between different test collections. To illustrate the influence of $T$ on gradual transition detection performance, we plotted a graph with recall, precision, and quality for the 2004 test collection with the optimal settings $W_g = 18$, $M = 1$ and varied values for $T$. The resulting graph is shown in Figure 3.7. As can be expected, large values for $T$ cause decreased recall and improved precision because only transitions that are associated with a high peak value in the $R_{pp}$ curve are detected, while all other transitions are missed. Similarly, small values for $T$ cause a low threshold that results in high recall but poor precision because many local maxima in the $R_{pp}$ curve will be falsely
Figure 3.7: Quality, recall, and precision when varying the threshold factor $T$ for gradual transition detection on the TRECVID 2004 test collection using the L-HSV feature.

reported as transitions. We thus observe a clear maximum in the quality curve — for this collection at $T = 1.6$ — that indicates the optimum value for $T$.

The results of our experiments show that, with the exception of cut detection on the 2006 test set, the localised (L-HSV) histogram feature allows more effective detection of shot boundaries with our algorithm. However, feature extraction and frame comparison are computationally more expensive for the localised histogram feature than for the global histograms. To quantify the difference in processing speed between both features, we conducted comparative timing experiments with our system. We use the **mpeg2decode** video decoder provided by the MPEG Software Simulation Group (MSSG)\(^3\) for decoding MPEG-1 compressed video. We altered this program to extract the desired histograms and to store the feature data in a binary file on disk. Our segmentation system operates independently on the binary feature data file. Both these implementations are single-threaded and not optimised for efficiency, and the results shown here are only for comparing the efficiency of the histogram features that we used in our experiments.

\(^3\)http://www.mpeg.org/MSSG
Chapter 3. Temporal Video Shot Segmentation

Table 3.4: Comparative timing results for the complete TRECVID 2004 test set using the global HSV histograms versus the localised HSV histograms. The table shows processing time in seconds and processing speed in frames per second (fps). The global histograms are generally processed faster, especially during segmentation. Overall, the difference is not as dramatic because decompressing the video takes up the largest part of the computation time.

<table>
<thead>
<tr>
<th></th>
<th>Global HSV histogram</th>
<th>Localised HSV histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>seconds</td>
<td>fps</td>
</tr>
<tr>
<td>Decoding &amp; feature extraction</td>
<td>12 872.2</td>
<td>48.0</td>
</tr>
<tr>
<td>Segmentation</td>
<td>10.9</td>
<td>56 877.4</td>
</tr>
<tr>
<td>Combined</td>
<td>12 883.1</td>
<td>48.0</td>
</tr>
</tbody>
</table>

We conducted the timing experiments on a single CPU computer with 1 GB RAM and an AMD-Athlon 64 bit processor, running at 2.2 GHz with a standard installation of openSUSE Linux 10.2 based on kernel 2.6.18. Table 3.4 shows the timing results for processing the complete TRECVID 2004 test collection using the global HSV histograms versus the localised HSV histograms. Results are shown as processing time in seconds and processing speed in frames per second (fps). The global HSV histogram feature is processed faster than the localised histograms; decoding the videos and extracting the 32 bin global HSV histograms for all twelve clips of the 2004 test collection took approximately 12 872 seconds (3 hours and 34.5 minutes). This equals 48 fps.

In contrast, video decoding and extracting 32 bin, 4 x 4 localised histograms takes approximately 15 147 seconds (4 hours and 12.4 minutes), which equates to a processing speed of only 41 fps. The difference during segmentation is more dramatic: our system processes the test set using global HSV histograms in under eleven seconds, while processing using L-HSV histograms requires over 210 seconds. The global histograms can be processed nearly twenty times faster than the localised histograms. However, as decoding the video and extracting the histograms takes between 70 to 1100 times longer than the actual segmentation step, the overall difference approximates the difference of the decoding and feature extraction process. As shown in Table 3.4, the overall processing speed using the global histogram feature was 48 frames per second on our system, versus approximately 40 frames per second when using the localised histogram feature. Considering the standard NTSC frame rate of all TRECVID test videos of 29.97 frames per second, this means that our system can process the videos at approximately 62% and 74% real time, respectively, on a standard desktop computer.
3.4. PERFORMANCE EVALUATION

3.4.3 Comparison with other Approaches

As part of our experiments in developing this algorithm, we participated in the TRECVID shot boundary evaluation tasks from 2003 to 2006. Tahaghoghi et al. applied their ranking approach in their participation in 2002, which we used as a basis for our cut detection stage. We compare our approach to those of other TRECVID participants for the years 2002 to 2006 by ranking all submitted runs of those years by overall Quality. Using Equation 2.16, we compute the Quality for each run based on Recall and Precision for all transitions. Table 3.5 shows the results for the best run of each year, the average over all submitted runs, and the worst result in comparison with our best results with the current algorithm. For the years 2003 to 2006, we also include results of our best blind run which was submitted to NIST as part of our TRECVID participation. The results of our current algorithm are obtained after parameter tuning, whereas the submitted runs are blind runs without prior knowledge of the test collection and parameter tuning on the test sets of previous years.

The first part of Table 3.5 shows the results for 2002. For this year, we include the best submitted run by Tahaghoghi et al. using the RWav feature and their best run using global HSV histograms. As we did not participate with our system in 2002, we only compare the TRECVID 2002 results to our current (best) result that uses the L-HSV feature. As can be seen from this table, the ranking approach by Tahaghoghi et al. performs well in detecting cuts with the wavelet feature, but is not useful for detecting gradual transitions. At the same time, while not performing best, this approach allows competitive results using the global HSV histogram feature. The best submitted run in 2002 used the IBM shot boundary detection system [Adams et al., 2002].

For 2003, Table 3.5 shows the performance of an early version of our algorithm in blind runs on the test set of that year. This version used the ranking approach for cut detection and our proposed average frame similarity method for gradual transition detection. In contrast to the current gradual transition detection algorithm, this version did not compare the last frame of previous shots with the first frame of new shots in addition to monitoring the $R_{pp}$ curve. Instead, we monitored the average sum of frame distances in the moving window of frames, and set a global threshold. In these runs, we used global HSV histograms with 192 bins per colour component. As can be seen from Table 3.5, we achieved competitive recall and good precision for cuts. Precision for gradual transitions was very high but only in combination with relatively poor recall. The best submitted run in 2003 used the IBM shot boundary detection system [Amir et al., 2003].
<table>
<thead>
<tr>
<th></th>
<th>All transitions</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>TRECVID 2002 runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best (overall Quality)</td>
<td>0.8840</td>
<td>0.8280</td>
<td>0.9340</td>
<td>0.8710</td>
</tr>
<tr>
<td>Average</td>
<td>0.7599</td>
<td>0.7897</td>
<td>0.8518</td>
<td>0.8354</td>
</tr>
<tr>
<td>Worst (overall Quality)</td>
<td>0.1780</td>
<td>0.3730</td>
<td>0.1800</td>
<td>0.4600</td>
</tr>
<tr>
<td>Tahaghoghi et al. (RWave)</td>
<td>0.6550</td>
<td>0.9400</td>
<td>0.9130</td>
<td>0.9420</td>
</tr>
<tr>
<td>Tahaghoghi et al. (HSV)</td>
<td>0.8060</td>
<td>0.7840</td>
<td>0.8420</td>
<td>0.9140</td>
</tr>
<tr>
<td>Our current algorithm</td>
<td>0.8840</td>
<td>0.8380</td>
<td>0.9330</td>
<td>0.9100</td>
</tr>
<tr>
<td>TRECVID 2003 runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best (overall Quality)</td>
<td>0.8990</td>
<td>0.9130</td>
<td>0.9380</td>
<td>0.9420</td>
</tr>
<tr>
<td>Average</td>
<td>0.7537</td>
<td>0.8225</td>
<td>0.8571</td>
<td>0.8685</td>
</tr>
<tr>
<td>Worst (overall Quality)</td>
<td>0.3220</td>
<td>0.7070</td>
<td>0.3540</td>
<td>0.9420</td>
</tr>
<tr>
<td>Our best submission (2003)</td>
<td>0.7920</td>
<td>0.8700</td>
<td>0.9180</td>
<td>0.8680</td>
</tr>
<tr>
<td>Our current algorithm</td>
<td>0.8870</td>
<td>0.8650</td>
<td>0.9200</td>
<td>0.9170</td>
</tr>
<tr>
<td>TRECVID 2004 runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best (overall Quality)</td>
<td>0.9030</td>
<td>0.8720</td>
<td>0.9290</td>
<td>0.9230</td>
</tr>
<tr>
<td>Average</td>
<td>0.7459</td>
<td>0.7487</td>
<td>0.8472</td>
<td>0.7851</td>
</tr>
<tr>
<td>Worst (overall Quality)</td>
<td>0.6120</td>
<td>0.1450</td>
<td>0.7340</td>
<td>0.1370</td>
</tr>
<tr>
<td>Our best submission (2004)</td>
<td>0.8810</td>
<td>0.8990</td>
<td>0.9440</td>
<td>0.9210</td>
</tr>
<tr>
<td>Our current algorithm</td>
<td>0.8990</td>
<td>0.8730</td>
<td>0.9300</td>
<td>0.9240</td>
</tr>
<tr>
<td>TRECVID 2005 runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best (overall Quality)</td>
<td>0.9250</td>
<td>0.8560</td>
<td>0.9490</td>
<td>0.9140</td>
</tr>
<tr>
<td>Average</td>
<td>0.7881</td>
<td>0.7328</td>
<td>0.8508</td>
<td>0.7856</td>
</tr>
<tr>
<td>Worst (overall Quality)</td>
<td>0.0970</td>
<td>0.1660</td>
<td>0.1240</td>
<td>0.1640</td>
</tr>
<tr>
<td>Our best submission (2005)</td>
<td>0.8910</td>
<td>0.8180</td>
<td>0.9170</td>
<td>0.9290</td>
</tr>
<tr>
<td>Our current algorithm</td>
<td>0.8940</td>
<td>0.8360</td>
<td>0.9450</td>
<td>0.9080</td>
</tr>
<tr>
<td>TRECVID 2006 runs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best (overall Quality)</td>
<td>0.8550</td>
<td>0.8920</td>
<td>0.8890</td>
<td>0.9040</td>
</tr>
<tr>
<td>Average</td>
<td>0.6679</td>
<td>0.6870</td>
<td>0.7290</td>
<td>0.7225</td>
</tr>
<tr>
<td>Worst (overall Quality)</td>
<td>0.4250</td>
<td>0.1430</td>
<td>0.4460</td>
<td>0.1220</td>
</tr>
<tr>
<td>Our best submission (2006)</td>
<td>0.8360</td>
<td>0.7380</td>
<td>0.8610</td>
<td>0.8740</td>
</tr>
<tr>
<td>Our current algorithm</td>
<td>0.8020</td>
<td>0.7450</td>
<td>0.8590</td>
<td>0.8280</td>
</tr>
</tbody>
</table>

Table 3.5: Results of our current algorithm on the TRECVID shot boundary test collections with optimal parameters in comparison to results of other participants. For the years 2003 to 2006, we include results on blind runs with our algorithm that were part of our participation at TRECVID.
3.4. PERFORMANCE EVALUATION

As Table 3.5 shows, the performance of the 2004 version of our algorithm in blind runs is very good. This version reflects our current version more closely, as it uses localised HSV histograms for both cut and gradual transition detection. Moreover, for gradual transition detection, the last frame of a previous shot is compared with the first frame of a new shot as a second criterion besides to monitoring the $R_{pp}$ curve. We still applied global, fixed thresholds for this comparison. In 2004, we achieved very strong results for both cut and gradual transition detection in these blind runs on the TRECVID 2004 test set. This run was ranked second among all 144 submitted runs based on Quality for all transitions; four of our ten submitted runs were ranked among the top seven. The best run in that year used the system from the Heinrich-Hertz-Institute [Petersohn, 2004]. Our current implementation yields better quality than the system that we used in 2004, mostly due to parameter optimisation and the automatic thresholding scheme that we now use for the second transition detection criterion. This threshold optimises the trade-off between Recall and Transition so that Quality is maximised.

Our best blind run on the TRECVID 2005 test set was achieved with the same algorithm as in 2004. In all other runs, we experimented with a two-pass implementation of our algorithm and with a three-dimensional histogram representation but could not observe any improvements. We included two runs using our 2004 implementation as a baseline; one of these performed best among all our submitted runs. Again, our current implementation performs slightly better due to optimised parameters and the automatic thresholding scheme. As previously discussed, we observe a drop in precision for gradual transition detection due to the large amount of short gradual transitions (35%) in this test collection that are not detected as effectively as longer transitions by our algorithm. Still, the overall performance is very close to the best submitted run and well above average. The best submitted run in 2005 used the system of Yuan et al. [2005].

The last part of Table 3.5 shows the performance of our algorithm in the best blind run on the TRECVID 2006 test set. The algorithm is again the same as in 2004, using the L-HSV feature and fixed thresholds for the second transition detection criterion. Interestingly, our submission performed better on this test collection with the fixed thresholds than our current version using automatic threshold computation. The short gradual transitions that constitute over 47% of this test collection cause a significant drop in cut and gradual transition detection precision. However, the overall performance of our algorithm is still well above the average performance of all other submissions for that year. The best submitted run that year was by the system developed at AT&T Research Labs [Liu et al., 2006].
CHAPTER 3. TEMPORAL VIDEO SHOT SEGMENTATION

3.5 Summary

In this chapter, we presented improvements to the ranking approach, originally proposed by Tahaghoghi et al. [2002], that uses a moving window for cut detection. We demonstrated that this approach can be effective using a localised HSV histogram feature, and that its applicability across different collections can be improved by replacing a fixed threshold scheme with a dynamic threshold. We also proposed a novel method for gradual transition detection that is based on average frame similarity within a window of consecutive frames. This method is highly effective using the same localised histogram feature that we proposed for cut detection. In most cases, this feature produces better results than global HSV histograms, but at the cost of increased computational complexity. Similar to cut detection, we apply a dynamic threshold when comparing frames bordering possible transitions as a second criterion, keeping the number of parameters that need to be set to a minimum.

Very short gradual transitions, have been shown to be problematic for our gradual transition detection stage. If all frames that are part of a gradual transition fit into one half-window, the peak in the $R_{pp}$ curve does not occur at the end of the transition but rather well after it. Our system falsely reports the end of a transition in such cases. Globally reducing the window size results in only limited success as recall decreases significantly for small window sizes. This problem could be addressed by monitoring a second $R_{pp}$ curve computed for a smaller part of the moving window in addition to the $R_{pp}$ curve for the full window. A second weakness is that our algorithm requires the collection-dependent constant threshold factor $T$ to be set. This reduces the ability of our algorithm to automatically adapt to different video collections. However, in experiments with several large test collections, we showed that our approach can be highly effective.

We integrate both cut and gradual transition detection as a single-pass algorithm that requires only one histogram feature for effective shot boundary detection, while most other effective approaches require the processing of several low-level features. We see this as a major benefit of our algorithm as it enables the implementation of an efficient and effective shot boundary detection system.
Chapter 4

Modelling Human Judgement of Digital Imagery

“There are no facts, only interpretations.”
– Notebooks by Friedrich Nietzsche

In the previous chapter, we discussed video shot segmentation of digital video as an important basis for handling footage effectively. A common step after shot segmentation is the semantic classification of shots to build an index. Low-level features, such as colour, texture, and shape are extracted for each shot to serve as a feature vector, or a feature descriptor. The critical step is then to classify the video segments according to their meaning — their semantics — in an automated process. This can be seen as generating a mapping from low-level features to high-level semantic concepts. For example, we could classify video shots whether they show a Car, a Person, or Snow. The terms Car, Person, and Snow are semantic high-level concepts.

Supervised learning techniques such as Bayesian networks or support-vector machines have been proven to perform well in automatic semantic classification of video [Over et al., 2005; Snoek and Worringer, 2005]. These methods require a training phase using manually classified examples to compute a binary model based on the low-level feature vectors of each example. After successful training, unseen video shots can be automatically separated into one of two classes: positive or negative, using the binary model. This is done for each concept so that positive refers to “can be described with the given concept” and negative refers to “can not be described with the given concept”. An important issue is to provide a sufficiently large number of correctly classified, discriminative examples for training as this
enhances the likelihood of resulting in good classification performance. Provided that models can be generated for very many concepts, such a model-driven approach promises to provide a generalised technique to video retrieval [Hauptmann et al., 2007; Snoek et al., 2005; 2007].

Providing high-quality manual annotations for large collections is, however, a challenging task. In particular, manual annotation is subjective and prone to error. As visual information almost always leaves room for individual interpretation and ambiguity, manually generated annotation cannot always be regarded as the one and only truth — there is no gold standard. Moreover, in an effort to annotate many items in a short time, human error may increase due to fatigue. Some form of quality assurance is needed to quantify and address this. A suitable way to alleviate the above problems is to obtain multiple annotations per video shot for a given concept. By assigning several raters to the task, and having each shot judged more than once, we expect an improvement in annotation quality because we can model the view of several raters, rather than only an individual opinion.

However, this introduces a new problem: we must now be able to handle disagreement between raters. It is problematic to decide whether a shot should be used as a positive or as a negative example if the ratings disagree. At the same time, inter-rater agreement is a good indicator for annotation quality. If we observe a high agreement rate, the annotation is likely to be accurate and reliable. But it is often not obvious how to quantify the inter-rater agreement because common methods, such as the well-known Kappa statistic [Cohen, 1960; Fleiss, 1981], may not be suitable for the given dataset.

In this chapter, we propose a Latent Class Modelling (LCM) [Lazarsfeld and Henry, 1968] approach to categorise video shots based on multiple non-uniform judgements. In Section 4.1, we describe a large-scale, collaborative annotation effort in which we obtained multiple judgements on video shots for 39 semantic concepts. We then demonstrate in Section 4.2 how latent class modelling can be used to clearly classify the individual video shots on the basis of the multiple, non-uniform ratings. We discuss the results of this modelling process in Section 4.3, and summarise our findings in Section 4.4.

4.1 Large-scale Annotation of Visual Documents

Labelling video shots in a controlled fashion is an effective approach to generating an annotated collection for training of supervised learning algorithms. As described in Section 2.3.2, the TRECVID annotation forum, first conducted in 2003, is an initiative by many of the regular TREC Video Retrieval Evaluation [Over et al., 2005; Smeaton et al., 2003] partici-
4.1. LARGE-SCALE ANNOTATION OF VISUAL DOCUMENTS

pants to generate large annotated training collections. We were involved in organising and conducting the second TRECVID annotation forum in 2005 in which the TRECVID 2005 development corpus was annotated. This corpus features approximately 80 hours of television news and entertainment recordings in 137 individual clips. These were temporally segmented into 61,904 distinct shots [Petersohn, 2004], and for each shot, a single representative frame was selected [Cooke et al., 2004]. More details on this collection are given in Appendix C.

The task was then to annotate each video shot by reviewing their representative frames. Each concept label that was assigned to a representative frame was also assigned to the full video shot. In practice, the annotation effort in 2005 was thus an image annotation task. Annotations were assigned globally, that is a concept label always applies to the entire frame and not only to a particular region of the frame.

During the organisation of this second annotation effort, participants agreed that it would be more useful to define a smaller vocabulary and aim for more annotations for each concept to promote consistency and accuracy. In particular, free-text labels were not allowed, and the annotation was conducted on a single key frame per shot to increase efficiency. Naphade et al. [2005] designed a concept lexicon of 44 semantic concept labels specifically for the purpose of annotating television broadcast footage. Five of the concepts were dropped from the vocabulary to be consistent with the requirement that all concepts can be annotated based on a still image. As a result, 39 of these concepts were finally selected to be used in the annotation effort in 2005. Details about all semantic concepts can be found in Appendix B. The Informedia Team\(^1\) at Carnegie-Mellon University, and the Intelligent Information Analysis group\(^2\) at the IBM T. J. Watson Research Center supported the effort by providing the annotation systems that were used. More than one hundred individual human annotators volunteered to participate, and each had the choice to use either one or both of the systems. The annotation task required that video key frames be classified into one of three possible categories in regards to each concept:

**positive:** The frame can clearly be classified as depicting the given concept, and should be used as a positive example;

**negative:** The frame can clearly be classified as not depicting the given concept, and should be used as a negative example;

\(^1\)http://www.informedia.cs.cmu.edu
\(^2\)http://www.research.ibm.com/iia
The frame should not be used as an example at all; users can assign this state to indicate that an image should neither be used as a positive nor as a negative example, including imperfect or blurred frames. This is the default state: All images that are not yet reviewed are labelled “skip”.

We presented an initial analysis of the data obtained with the IBM system [Volkmer et al., 2005a] which included up to four independent judgements for each concept. However, this did not include an analysis on the complete data, that is the data that was finally published for use in the subsequent TRECVID experiments and evaluations. The complete dataset consists of all annotations that were collected using the IBM and the CMU annotation systems, and includes up to five annotations per image. One of the key questions, how to model multiple user responses, was left unanswered. In the remainder of this chapter, we present a solution to this problem using latent class modelling.

### 4.2 Modelling Multiple Judgements

The manual annotation strategy, as specified above, results in multinomial ratings with three possible categories that were obtained for each image in regards to each concept. The large user base allowed multiple judgements to be acquired for a substantial part of the corpus, with up to five ratings for each image.

We omit images with one or more “skip” judgements. In preliminary evaluation experiments in which we tested different latent class models, such as those that include “skip” ratings, we could not achieve good model fit. We have thus selected an evaluation using a two-class latent class model as described here. Moreover, we believe that this greatly improves the overall quality of the training collection, while only slightly reducing the collection size. The multinomial ratings are thus reduced to binomial ratings; each remaining annotation indicates an annotator’s opinion that a given image is either a positive example or a negative example of a given concept.

In Figure 4.1, we illustrate the number of positively and negatively rated example images among all images in the collection with only a single rating. The grey bars represent the number of images rated “negative”, while the black bars represent the number of images rated “positive” for each concept in this group. Figure 4.2 shows the positive ratings for all images with two ratings. The graph in Figure 4.2 does not show images without any positive
4.2. MODELLING MULTIPLE JUDGEMENTS

Figure 4.1: Positive and negative ratings for the image/concept pairs with a single judgement. The yellow bars represent the number of images rated “negative”, while the blue bars represent the number of images rated “positive”. The graph shows only 30 concepts because there were no images with only a single rating for nine of the 39 concepts.

ratings, that is those that were unanimously rated “negative”. The number of these images is generally large compared to those with positive ratings, the scaling of the graph would thus make it difficult to identify details of the positive ratings. Similarly, the graph in Figure 4.3 shows the positive ratings for all images with three ratings. Figure 4.4 and Figure 4.5 show the positive ratings for the images with four and five ratings, respectively. As can be deduced from these graphs, not all images and concepts were reviewed by the same number of users. For example, within some concepts there are no images with only one rating because each image has been at least reviewed by two different annotators for the given concept. As shown in Figure 4.5, we observe the maximum of five independent ratings only for a small number of images in regards to five semantic concepts.

However, it is important for our approach that there is a substantial number of images with at least two independent ratings for all concepts. Otherwise, the model that we apply
Figure 4.2: Positive ratings for the image/concept pairs with two judgements. The blue bars represent the number of images that were rated positive in agreement by both raters while the yellow bars illustrate the number of images that were judged with disagreement.

would not be identifiable or achieve only poor model fit. We discuss model fit and model identifiability in more detail later.

The graphs in Figure 4.1 to Figure 4.5 also illustrate the underlying problem with diverging rater responses as they clearly show the substantial disagreement between ratings that we observe for all concepts. We consider the concept Bus as an example to elaborate on the problem; Figure 4.6 shows all ratings that we obtained this for concept. As we explained below in more detail, we group images within each concept depending on how many ratings are observed for each. We refer to these groups as subgroups. Subgroup 1 contains all images with one rating, subgroup 2 contains all images with two ratings and so on. Each row in Figure 4.6 represents the ratings for one subgroup. The columns in the centre of this table show the number of images for which we observed the given number of positive ratings. For example, for subgroup 5, all five raters agreed on 17,946 images that these were negative examples for the concept Bus. The five raters were also in agreement that there were four
images that are a positive example. However, 34 images were judged to be positive by only one rater, while the other four raters disagreed. Similarly, 37 images were judged positive by two raters, 15 images were judged positive by three raters, and 9 images were judged positive by four raters. Thus, in the subgroup of images with five ratings, we have 99 potential examples for Bus.

The problem is now to classify each image based on these ratings. A naïve approach could be to use a simple averaging or voting scheme, but this will fail in case there is a tie in the ratings. More importantly, this would neglect differences in the response behaviour across different concepts. On the basis of our experience, we also expect there is a greater likelihood that a user will overlook an existing object rather than claiming to have seen a non-existent object; in other words, false negatives are more likely than false positives. Moreover, the concepts have very different distributions; some are very common, while others are very rare. This in turn leads to different error rates, and consequently to different ratios between false positives and false negatives. One could revert to using only unanimously rated images, but...
this is not desirable as it is unlikely to yield enough positive examples for rare concepts. Considering the ratings for subgroup 5 in Figure 4.6, we would neglect 95 possible training examples and only make use of four images. Among all images in this table, there are only 20 cases in which all raters agreed; the baseline approach would thus only yield 20 positive training examples. However, it is highly likely that there are more positive examples among those that were rated as positive at least once. Aside from this, using only unanimously judged images would render most of the effort spent during annotation inconsequential.

4.2.1 Latent Class Modelling

Latent class modelling provides an adequate solution for classifying images based on multiple disagreeing ratings. Moreover, it allows us to use varying numbers of ratings for each image. One central assumption of the latent class model is that the actual classification of an image is contained in our annotation only as a latent variable $X$ [Vermunt and Magidson, 2003],
4.2. MODELLING MULTIPLE JUDGEMENTS

Figure 4.5: Positive ratings for the image/concept pairs with five judgements. Interestingly, for three out of the five given concepts, no image was rated unanimously as a positive example.

that is we cannot observe it directly. However, we can derive $X$ from a number of observable (manifest) variables. The manifest variables are the ratings that we observe for each image; we combine these ratings into a response vector $\mathbf{Y} = [Y_1, \ldots, Y_K]$. In our case, the response vector $\mathbf{Y}$ may have up to five components and we define $K \in \{1, \ldots, 5\}$ as the number of observed ratings for a given image. A specific rater’s response $y$ can have one of the two values 0 (negative) and 1 (positive). Consequently, we assume the latent variable $X$ to be dichotomous, that is it can represent either one of the two latent classes $x = 0$ (negative) and $x = 1$ (positive). The number of latent classes for our model is therefore $C = 2$. In the latent class modelling approach, the probability of obtaining a specific response vector for a given image is defined as the product of the individual response probabilities for each rater [Vermunt and Magidson, 2003]. For a given image that belongs to the latent class $x$,

\footnote{Statistical literature frequently uses the values 1 and 2 to express negative and positive ratings. However, we use 0 and 1, common in computer science, and we modify all our derived equations accordingly.}
we can therefore define the conditional response probability $P_C(\vec{Y} = \vec{y}|X = x)$ as follows:

$$P_C(\vec{Y} = \vec{y}|X = x) = \prod_{k=1}^{K} P(Y_k = y_k|X = x)$$  \hspace{1cm} (4.1)

An important assumption of this equation is local independence, that is the individual responses $y_k$ are mutually independent events. Our experiment setup satisfies the requirement of local independence because all raters judge images independently: a particular judgement does not have any influence on the judgement of another rater. Equation 4.1 defines the class-specific probability of obtaining the response $P(\vec{Y} = \vec{y})$. The unconditional response probability is defined as follows [Vermunt and Magidson, 2003]:

$$P_G(\vec{Y} = \vec{y}) = \sum_{x=1}^{C-1} P(X = x)P(\vec{Y} = \vec{y}|X = x)$$  \hspace{1cm} (4.2)

where $P(X = x)$ is the proportion of images that belong to class $x$. In Equation 4.2, it is valid to build the sum of the class-specific probabilities because the latent classes are mutually exclusive. This reflects that, by definition, an image can be either a positive or a negative example for a given concept. While this promotes consistency, it also limits the ability of reflecting the reality accurately. For example, annotators may wish to distinguish between more and less suitable examples for a given concept. This is not possible with the current design of the annotation process. As we will describe below, however, latent class modelling allows us to make a statement about the suitability of specific images based on their posterior class membership probabilities after the modelling process. As can be seen

\[
\begin{array}{c|ccccc|c}
\text{Subgroup}\hspace{1cm} (\# \text{ ratings}) & \text{0} & \text{1} & \text{2} & \text{3} & \text{4} & \text{Potential no. of positives} \\
\hline
5 & 17,946 & 34 & 37 & 15 & 9 & 499 \\
4 & 22,709 & 41 & 17 & 14 & 14 & 86 \\
3 & 20,373 & 47 & 14 & 2 & 14 & 63 \\
2 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Positive classified examples (unanimously rated)} & 29 \\
\text{Positive classified examples (LCM, modal classification)} & 173
\end{array}
\]
4.2. MODELLING MULTIPLE JUDGEMENTS

from Equation 4.2, the probability $P_G(\vec{Y} = \vec{y})$ is a weighted average of the class-specific probabilities $P_G(\vec{Y} = \vec{y}|X = x)$ over $C$ classes. The combination of Equation 4.1 and Equation 4.2 yields the definition of the latent class model that can be used to estimate the conditional response probabilities $P(Y_k = y_k|X = x)$ if $K$ is constant [Vermunt and Magidson, 2003]:

$$P_G(\vec{Y} = \vec{y}) = \sum_{x=0}^{C-1} \left( P(X = x) \prod_{k=1}^{K} P(Y_k = y_k|X = x) \right)$$ (4.3)

This model is also referred to as the general latent class model. However, this general model is not applicable in our case because the number of ratings $K$ per image is not constant in the TRECVID 2005 annotation data. For some images, we have only one observed rating, while we observed up to five ratings for other images. As Vermunt [1996; 1997] describes, response vectors of varying size can be handled by defining subgroups for each observed vector length. By introducing a response indicator that implies the subgroup to which an image belongs, we make the model applicable for partially missing-data due to nonresponse. This means that we have no response at all by some raters for a part of the collection.

We apply this strategy to our data, and group all images into subgroups depending on the number of ratings that we observe for each. As the maximum is five ratings, we define five subgroups and introduce the response indicator $R = r$, with $r \in \{1, \ldots, m\}$, where $m = 5$. The number of ratings per image is constant within each subgroup, and so Equation 4.3 can be reformulated to describe the response probability for a given subgroup $R = r$ by replacing $K$ with the number of responses $r$ for that subgroup:

$$P_{CS}(\vec{Y} = \vec{y}|R = r) = \sum_{x=0}^{C-1} \left( P(X = x) \prod_{k=1}^{r} P(Y_k = y_k|X = x) \right)$$ (4.4)

An image in our collection can belong to only one subgroup, that is the subgroups are mutually exclusive. Similar to the step that led us from Equation 4.2 to Equation 4.3, we can therefore assume that the unconditional response probability over all subgroups is expressed as a weighted average of all subgroup-specific probabilities $P(\vec{Y} = \vec{y}|R = r)$. This can be formulated as follows:

$$P_{GS}(\vec{Y} = \vec{y}) = \sum_{r=1}^{m} P(R = r) P(\vec{Y} = \vec{y}|R = r)$$ (4.5)

To arrive at a model that allows us to handle the varying numbers of judgements in our data, we combine Equation 4.4 with Equation 4.5. This yields the latent class model that
we finally use in our estimation process; it is described by the following equation:

\[
P_{\text{GS}}(\vec{Y} = \vec{y}) = \sum_{r=1}^{m} P(R = r) \left\{ \sum_{x=0}^{C-1} \left( P(X = x) \prod_{k=1}^{r} P(Y_k = y_k \mid X = x) \right) \right\}
\]

(4.6)

So far, we have neglected the requirement of latent class modelling that observations are not interchangeable. The latent class models in above equations include the assumption that a specific response \( y_k, k \in \{1, \ldots, K\} \), originates from the same rater for all images in the collection. This is not given in the TRECVID annotation setup as the ratings of a particular image can originate from any of the over 100 raters. To apply latent class modelling to our data, we must impose restrictions on the model during the estimation process to handle the interchangeability of ratings in our data. Specifically, we force equality of the conditional response probabilities by specifying:

\[
P(Y_1 = y_1 \mid X = x) = P(Y_2 = y_2 \mid X = x) = \cdots = P(Y_K = y_K \mid X = x)
\]

This means that all raters are equally likely to make a particular judgement, given a specific image. While this is a tremendous simplification of the reality, it is a requirement that we must enforce due to the nature of the TRECVID data. The originally unrestricted latent class model is thus transferred into a restricted latent class model with two latent classes. Aside from correctly modelling the rater interchangeability, the restriction reduces the degrees of freedom of our system. An unrestricted latent class model requires at least three manifest variables for estimating a model with two latent classes [Vermunt and Magidson, 2003]. Due to the restriction that we applied, we require only two observed ratings per image to estimate the model. The subgrouping has a similarly beneficial effect: as we estimate the response behaviour of all subgroups within the same model, we can include the subgroup with one rating as long as there are other subgroups with two or more ratings per image for the same concept. This is the case for all concepts in the TRECVID 2005 annotation data.

The model and its parameters can be estimated with any Expectation Maximisation (EM) algorithm [Dempster et al., 1977; Hartley, 1958]. We used the \( \ell \text{EM} \) program [Vermunt, 1997] to estimate our model. The output of the estimation process contains the latent class probabilities \( P(X = x) \) for the two latent classes and the individual response probabilities \( P(\vec{Y} = \vec{y}) \). The latent class probabilities \( P(X = x) \) for \( x \in \{0, 1\} \) allow us to estimate the expected frequency of each concept in the collection, based on the model prediction. For example, \( P(X = 1) \) is the probability of an image being a positive example for the given concept if the image were picked randomly out of the collection. To classify a particular
4.2. MODELLING MULTIPLE JUDGEMENTS

Figure 4.7: The expected concept prevalences based on the model prediction (light blue bars) in comparison with the number of positive examples obtained by modal classification (dark blue bars). While both generally correlate well, we observe some outliers, for example the concepts Police/Security, Desert, Corporate Leader, Office, and Vegetation.

image into one of the two latent classes based on the observed ratings, we apply the following Bayesian rule [Vermunt and Magidson, 2003]:

$$P(X = x | \vec{Y} = \vec{y}) = \frac{P(X = x)P(\vec{Y} = \vec{y} | X = x)}{P(\vec{Y} = \vec{y})} \tag{4.7}$$

We use modal classification to assign an individual image to a particular class; each image is assigned to the class with the highest probability $P(X = x | \vec{Y} = \vec{y})$. This enables us to clearly classify each image in the collection, and finally to generate the training collection for each concept. We expect that the number of images obtained with modal classification to be close to the number of examples predicted by the latent class probability. Figure 4.7 shows both statistics as a bar graph, ordered by concept frequency. The grey bars represent the model prediction in terms of expected positive example images, while the black bars represent the number of positive examples obtained by modal classification. We observe a generally strong correlation between both; however, for the concepts Police/Security, Desert, Corporate Leader, Office, and Vegetation the prediction differs considerably from the actual
number of examples obtained. This is an indicator for relatively high levels of disagreement in the judgements for these concepts.

We also notice substantial differences in the concept frequencies. The most frequent concepts are *Person* and *Face* with 39,889 shots and 24,222 shots, respectively. In contrast, according to the observed ratings only 67 shots are classified as *Charts* and 84 shots are classified as *Natural-Disaster*.\(^5\) For the concept *Bus*, as illustrated in Figure 4.6, we obtain 173 positive examples using LCM-based modal classification — a substantial improvement over 20 examples when only using unanimously judged images. The exact numbers for all concepts can be found in Table 4.2 on page 94.

### 4.2.2 Estimating Annotation Quality

An important consideration when modelling annotations is the quality of the annotations themselves. Often, the rate of agreement between judgements is used as a measure of the reliability of the ratings [Fleiss, 1981]. The well-known Kappa statistic by Cohen [1960], or the more general formulation of Kappa by Fleiss [1981] are frequently used for such purposes. While Cohen’s Kappa is limited to the comparison of two raters, Fleiss’ Kappa is applicable to more than two raters and is sometimes referred to as the *intraclass correlation coefficient*. Both Cohen’s and Fleiss’ Kappa are commonly denoted by \(\kappa\), and based on the following formula [Fleiss, 1981]:

\[
\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \tag{4.8}
\]

where \(\overline{P}\) and \(\overline{P}_e\) are defined as:

\[
\overline{P} = \frac{1}{Nr(r-1)} \left( \sum_{i=1}^{N} \sum_{j=0}^{C-1} n_{ij}^2 - Nr \right) \quad \text{and} \quad \overline{P}_e = \sum_{j=0}^{C-1} p_j^2 \quad \text{with} \quad p_j = \frac{1}{Nr} \sum_{i=1}^{N} n_{ij}
\]

and \(N\) is the sample size, that is the number of images that have been rated for the given concept. \(C\) denotes the number different categories, which in our case equals the number of latent classes. Further, \(n_{ij}\) denotes the number of judges who assigned the \(i\)th image to the \(j\)th class. The number of observed ratings per image is denoted by \(r\). This implies that, for a dataset such as ours, \(\kappa\) is calculated per subgroup and the final \(\kappa\) value is the result of the weighted average over all subgroups.

---

\(^5\) Differences to previously published results [Volkmer et al., 2005a] are due to the more accurate evaluation method we have used here and because we have incorporated more ratings into estimating the latent class model.
4.2. MODELLING MULTIPLE JUDGEMENTS

The Kappa statistic reports the inter-rater agreement as a value between $-1$ and $1$. Perfect agreement is indicated for $\kappa = 1$, while $\kappa = 0$ stands for the level of agreement that is expected from random assignment, thus called agreement by chance. Negative values of $\kappa$ indicate agreement below that expected from random assignment. The Kappa statistic is the subject of controversy in the literature: it is reported to be biased by trait prevalence, and to yield misleading results when the distribution of categories is skewed [DiEugenio and Glass, 2004; Uebersax, 1988]. As the results in Figure 4.7 show, this is in fact the case with our data. The Kappa statistic is thus likely to yield unreliable results.

Latent class modelling offers a different way of assessing annotation quality that is based on the model classification error $E$. The classification error $E$ is the estimated proportion of false classifications based on all classifications for a particular concept. When using the modal classification rule of Equation 4.7, we estimate the classification error $E$ for $N$ rated images as follows [Vermunt and Magidson, 2003]:

$$E = \sum_{i=1}^{I} n_i \left\{ 1 - \max \{ P(X = x \mid \hat{Y} = \hat{y}_i) \} \right\}$$

where $I$ denotes the possible number of different response patterns, and $n_i$ is the observed frequency for a particular response pattern. We then compare the LCM-based classification error $E$ to the proportion of classification errors based on the unconditional latent class probabilities $P(X = x)$. This results in the classification performance measure $\lambda$, that is defined as follows [Vermunt and Magidson, 2003]:

$$\lambda = 1 - \frac{E}{\max P(X = x)}$$

The classification performance $\lambda$ is not a measure of inter-rater agreement, but it estimates how well the class assignment can be performed using the observed ratings. It is well suited to our purposes as we are not actually interested in the disagreement itself, but rather in how well we are able to classify our examples. The LCM classification performance index $\lambda$ should be interpreted similar to an $R^2$ measure [Vermunt and Magidson, 2003]; it will approach a value of 1 for perfect classification.

To put the classification performance index into context, we compare $\lambda$ to the inter-rater agreement quantified with Fleiss’ Kappa in Figure 4.8. In this graph, the concepts are ordered by prevalence, with the most prevalent concept on the right. We observe a strong correlation between $\kappa$ and $\lambda$; the Pearson’s product-moment correlation coefficient is $\rho = 0.83$. However, we identify some outliers, in particular among the rare concepts, such
Figure 4.8: LCM classification performance index \( \lambda \) compared to the Fleiss’ Kappa \( \kappa \) for each concept. Low levels of agreement and poor classification performance are indicated for the concepts Police/Security, Desert, Corporate Leader, Office, and Vegetation.

as Natural-Disaster or Charts. For these concepts, the differences between \( \lambda \) and \( \kappa \) are more significant. We believe that the disagreement measure \( \kappa \) may not be reliable for concepts with very few positive judgements because of the possible bias of Kappa by the trait prevalence. According to the LCM-based classification performance measure \( \lambda \), the images could still be classified adequately for the concepts Natural-Disaster and Charts despite the disagreement. The fact that \( \lambda \) is not close to 1 for these concepts means that we have to expect some false classifications, but we believe that the annotation quality in these cases is not as poor as the low values of \( \kappa \) may suggest. In contrast, for the concepts Police/Security, Desert, Corporate Leader, Office, and Vegetation \( \lambda \) and \( \kappa \) are consistently low. This indicates that the level of disagreement is too high, and that the latent class model cannot classify these images well. As a result, the quality of the trained model may suffer.
4.3 Discussion of Results

4.2.3 Assessing Goodness-of-fit

After estimating the latent class model, it is important to assess goodness of fit to confirm whether we have chosen a suitable model. For each concept, our model assumes each image to belong to one of the two latent classes “positive” or “negative”. The goodness-of-fit test is in principle a test for statistical significance and helps to confirm that our assumption is indeed true. We do not expect the estimated ratings to be significantly different from the observed ratings. Goodness of fit is confirmed if this is indeed the case. In our case, we apply the likelihood-ratio-based Chi-Square test [Sokal and Rohlf, 1995] to the response tables, to compare the observed ratings with those estimated by the model. This test is also referred to as the G-Test; it ignores response patterns that are not observed and is therefore robust even for sparse response tables, for which Pearson’s Chi-Square test is likely to fail. In Table 4.1, we illustrate a goodness-of-fit analysis representatively for the concept Studio. The table shows the observed and the estimated frequencies for all subgroups with non-zero estimated results. The G-Test values are calculated per subgroup using the following formula [Sokal and Rohlf, 1995]:

\[ G = 2 \sum_i O_i \ln \left( \frac{O_i}{E_i} \right) \quad (4.11) \]

where \( O_i \) are the observed frequencies and \( E_i \) are the estimated frequencies. \( I \) is the number of observed response patterns.

The G-Test values for each subgroup are accumulated to result in the total value of the G-Test value for that concept. With the total degrees of freedom for the system \( v \), here \( v = 8 \) for the concept Studio, we determine the p-Value based on the Chi-Square distribution, in this case \( p = 0.002 \). We require the significance level to be \( \alpha = 0.01 \) (1%), and thus confirm good model fit for the concept Studio because \( p < 0.01 \). Using the likelihood-ratio Chi-Square test in the same way, we identified good fit of our two-class model for all concepts at the 1%-level.

4.3 Discussion of Results

In Figure 4.9, we show sample images of the TRECVID 2005 development collection that we classified for the concept Bus using LCM-based modal classification. As we use a two-class model, modal classification is equivalent to applying a threshold to the posterior class membership probability of 0.5. To illustrate the effect of the modelling process, we selected some interesting cases with a posterior class membership probability for the class positive
Table 4.1: An evaluation of the goodness of fit, representatively for the concept Studio. We observe fourteen different response patterns with non-zero estimated occurrences across three subgroups. The G-Test statistic over all subgroups is $G = 23.96$. With eight degrees of freedom, this results in a $p$-value of $p = 0.002$. Given our 1%-confidence level ($\alpha = 0.01$), we confirm good model fit.

above and below this threshold. For example, an image with two positive ratings out of five is still classified as a positive example for the concept Bus with a relatively high probability of $P(X = 1|\vec{Y} = \vec{y}) = 0.9991$, while an image with one positive rating out of four is classified “negative” with the class membership probability of $P(X = 1|\vec{Y} = \vec{y}) = 0.4150$. This may be different for other concepts because the model prediction is fitted to the response behaviour for each individual semantic concept, but it shows that this approach goes beyond simple averaging or voting. Latent class modelling incorporates multiple disagreeing ratings of a sampled population into the classification process, and models the response behaviour of this population.

We believe that this reflects a more accurate, generic view on the images and concepts used in the annotation. Using modal classification, we can compute posterior class membership probabilities for each image for unambiguous classification and obtain a maximum number of usable training examples. Compared to using only unanimously rated images we achieve substantial improvements in the number of usable examples. We illustrate the number of positive examples classified with latent class modelling in comparison to the number
4.3. DISCUSSION OF RESULTS

Figure 4.9: Selected examples for the concept Bus, classified using modal classification after the latent class modelling process, along with the computed posterior class membership probabilities in regards to the class positive.
Figure 4.10: The number of positive examples obtained with the baseline approach of using only unanimously rated images (light blue bars) compared to the number of positive examples obtained through latent class modelling (dark blue bars). In most cases, latent class modelling leads to more positive examples.

of unanimously rated images in Figure 4.10. For most concepts, latent class modelling yields more positive examples than the baseline approach of using only the images that have been rated in agreement. For concepts such as Urban, Vegetation, or Walking/Running we observe a large increase of positive examples. Detailed comparative results are shown in Table 4.2. From this table, it can be seen that the number of positive examples for concept Vegetation increases by approximately 2.6 times. For the concept Corporate Leader, the number of positive examples increased to more than twenty times the number that would be obtained when only using unanimously rated images.

Latent class modelling can also maximise the number of negative examples. This is shown in Figure 4.11 in which we compare the additional number of positive examples to the additional number of negative examples. We observe an increase of either positive or negative examples over the baseline approach for all concepts. In a few cases, such as Corporate Leader, or Entertainment, we observe an increase for both positive and negative examples.
4.3. DISCUSSION OF RESULTS

<table>
<thead>
<tr>
<th>Concept</th>
<th>LCM estimates</th>
<th>LCM classification</th>
<th>Baseline approach</th>
<th>Total # images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>Negative</td>
<td>Positive</td>
<td>Negative</td>
</tr>
<tr>
<td>Charts</td>
<td>161</td>
<td>61636</td>
<td>67</td>
<td>61730</td>
</tr>
<tr>
<td>Natural-Disaster</td>
<td>161</td>
<td>61657</td>
<td>84</td>
<td>61734</td>
</tr>
<tr>
<td>Prisoner</td>
<td>241</td>
<td>61555</td>
<td>130</td>
<td>61666</td>
</tr>
<tr>
<td>Bus</td>
<td>196</td>
<td>61081</td>
<td>173</td>
<td>61104</td>
</tr>
<tr>
<td>Boat/Ship</td>
<td>235</td>
<td>61542</td>
<td>196</td>
<td>61581</td>
</tr>
<tr>
<td>Court</td>
<td>309</td>
<td>61413</td>
<td>201</td>
<td>61521</td>
</tr>
<tr>
<td>Airplane</td>
<td>278</td>
<td>61481</td>
<td>229</td>
<td>61530</td>
</tr>
<tr>
<td>Mountain</td>
<td>426</td>
<td>61335</td>
<td>264</td>
<td>61497</td>
</tr>
<tr>
<td>People-Marching</td>
<td>574</td>
<td>61155</td>
<td>275</td>
<td>61454</td>
</tr>
<tr>
<td>Snow</td>
<td>379</td>
<td>60744</td>
<td>305</td>
<td>60818</td>
</tr>
<tr>
<td>Truck</td>
<td>474</td>
<td>61090</td>
<td>332</td>
<td>61232</td>
</tr>
<tr>
<td>Maps</td>
<td>417</td>
<td>60852</td>
<td>391</td>
<td>60878</td>
</tr>
<tr>
<td>Flag-US</td>
<td>1247</td>
<td>60204</td>
<td>474</td>
<td>60977</td>
</tr>
<tr>
<td>Animal</td>
<td>556</td>
<td>61213</td>
<td>504</td>
<td>61265</td>
</tr>
<tr>
<td>Police/Security</td>
<td>5320</td>
<td>56183</td>
<td>534</td>
<td>60969</td>
</tr>
<tr>
<td>Desert</td>
<td>1983</td>
<td>59084</td>
<td>544</td>
<td>61243</td>
</tr>
<tr>
<td>Weather</td>
<td>543</td>
<td>61163</td>
<td>588</td>
<td>61118</td>
</tr>
<tr>
<td>Explosion/Fire</td>
<td>794</td>
<td>60782</td>
<td>751</td>
<td>60825</td>
</tr>
<tr>
<td>Corporate Leader</td>
<td>2799</td>
<td>57782</td>
<td>1024</td>
<td>59557</td>
</tr>
<tr>
<td>Waterscape/Waterfront</td>
<td>1068</td>
<td>60638</td>
<td>1064</td>
<td>60642</td>
</tr>
<tr>
<td>Office</td>
<td>2667</td>
<td>57944</td>
<td>1231</td>
<td>59380</td>
</tr>
<tr>
<td>Sports</td>
<td>1595</td>
<td>59767</td>
<td>1406</td>
<td>59956</td>
</tr>
<tr>
<td>Government Leader</td>
<td>2949</td>
<td>57612</td>
<td>1867</td>
<td>58694</td>
</tr>
<tr>
<td>Road</td>
<td>2549</td>
<td>57986</td>
<td>2113</td>
<td>58422</td>
</tr>
<tr>
<td>Computer/TV-screen</td>
<td>3675</td>
<td>58089</td>
<td>2316</td>
<td>59448</td>
</tr>
<tr>
<td>Military</td>
<td>2240</td>
<td>59298</td>
<td>2325</td>
<td>59213</td>
</tr>
<tr>
<td>Meeting</td>
<td>2901</td>
<td>56673</td>
<td>2898</td>
<td>56676</td>
</tr>
<tr>
<td>Car</td>
<td>3004</td>
<td>57934</td>
<td>3061</td>
<td>57877</td>
</tr>
<tr>
<td>Sky</td>
<td>5212</td>
<td>56178</td>
<td>4244</td>
<td>56146</td>
</tr>
<tr>
<td>Urban</td>
<td>5652</td>
<td>49919</td>
<td>4401</td>
<td>51170</td>
</tr>
<tr>
<td>Vegetation</td>
<td>12544</td>
<td>49065</td>
<td>5205</td>
<td>56404</td>
</tr>
<tr>
<td>Walking/Running</td>
<td>4951</td>
<td>56092</td>
<td>5557</td>
<td>55486</td>
</tr>
<tr>
<td>Building</td>
<td>9202</td>
<td>52145</td>
<td>6044</td>
<td>55303</td>
</tr>
<tr>
<td>Entertainment</td>
<td>6814</td>
<td>48769</td>
<td>6099</td>
<td>49484</td>
</tr>
<tr>
<td>Studio</td>
<td>6817</td>
<td>54101</td>
<td>6245</td>
<td>54673</td>
</tr>
<tr>
<td>Crowd</td>
<td>8391</td>
<td>52460</td>
<td>7964</td>
<td>52887</td>
</tr>
<tr>
<td>Outdoor</td>
<td>17116</td>
<td>39785</td>
<td>17515</td>
<td>39386</td>
</tr>
<tr>
<td>Face</td>
<td>27062</td>
<td>34081</td>
<td>24222</td>
<td>36921</td>
</tr>
<tr>
<td>Person</td>
<td>40698</td>
<td>18820</td>
<td>39889</td>
<td>19629</td>
</tr>
</tbody>
</table>

Table 4.2: Final LCM-based results in comparison with the baseline approach. Columns 2 and 3 show the numbers of examples as estimated by the latent class model. Columns 4 and 5 show the numbers of examples as classified with the modal classification rule from Equation 4.7, these are the examples usable for training. Columns 6 and 7 show the numbers of examples obtained with the baseline approach.
The latent class modelling approach is mainly a way for resolving the ambiguity that arises from obtaining multiple judgements. We expect most additional examples to be gained for the concepts where there is a high level of disagreement ratings. To confirm this, we computed the ratio between positive examples identified by LCM and unanimously rated images. We plot this ratio against the classification performance $\lambda$ for all concepts in Figure 4.12, ordering all concepts by the gain ratio such that the concept with the highest ratio, Corporate Leader, is shown on the right. An inverse correlation can be observed when comparing $\lambda$ of the individual concepts to the gain ratios. That is, the increase in the number of examples when using latent class modelling tends to be higher when the classification performance is low for a concept. On the contrary, this means that latent class modelling maximises quantity at the expense of quality.

Considering, for example, the concept Corporate Leader, the baseline approach yields only 47 unanimously rated positive examples. For these 47 images, we have a high confidence in their correct classification. Latent class modelling classifies a total of 1024 images as positive in regards to the concept Corporate Leader, as shown in Table 4.2. While this means that we can train a semantic model on the basis of a substantial number of examples, it may also mean that we include many false positives into the training. This effect — if it is not addressed — is likely to lead to poor model quality and reduced classification performance.
4.3. DISCUSSION OF RESULTS

Figure 4.11: The number of positive and negative examples gained by using latent class modelling when compared to the baseline approach of using unanimously rated images. For some concepts the numbers of both positive and negative examples increase.

We can use the classification performance index $\lambda$ to identify problematic concepts and avoid this problem. Instead of using modal classification for concepts with low $\lambda$-values, a higher threshold can be applied to the class membership probability. This way, only examples that have been classified with higher agreement would be included in the training. An assessment of the classification performance using semantic models that have been trained with examples from a latent class modelling process is required to gain more insight. This will help to determine useful thresholds for $\lambda$ and the posterior class membership probabilities for each concept. In addition, the question about whether the class membership probabilities can be effectively incorporated into the training algorithms could be addressed.

Another observation is that we probably have far more positive examples for the most prevalent concepts such as Person or Face than needed to create a good model. This implies that we could optimise manpower in collaborative annotation efforts by stopping annotation of a concept as soon as enough positive examples have been identified. In addition, this
Figure 4.12: The ratio between positive examples identified by LCM and unanimously rated images compared to the classification performance $\lambda$. We observe an inverse correlation between both, that is latent class modelling maximises the quantity of example images at the expense of quality.

could be exploited such that we use only the best examples to build the semantic model by applying a threshold higher than 0.5 to the estimated posterior class membership probability. This could possibly improve model quality as it would reduce the likelihood of including false positives.

### 4.4 Summary

In this chapter, we discussed a latent class modelling approach for modelling human judgement of digital imagery for multimedia retrieval. This approach allows multiple non-uniform judgements for each image to be combined, and posterior class membership probabilities to be estimated so that all images can clearly be classified despite disagreement between raters. By helping to resolve ambiguity, the latent class modelling approach permits all ratings to be incorporated into building semantic models. It maximises the number of examples that can be used in training supervised learning algorithms. Particularly for very infrequent con-
4.4. SUMMARY

cepts, this can help to build more effective semantic models. We believe that using multiple annotations is preferable over using only single ratings because it allows a more generic view of visual content to be modelled.

The classification performance measure $\lambda$ allows us to assess how suitable the annotations are for computing a discriminative model. This does not allow us to conclude that a high score for $\lambda$ results in a high detection performance of a supervised learning method. The detection performance of a supervised classifier as such depends rather on the low-level features that are used but a high score for $\lambda$ means that the examples were annotated very consistently, and that the examples form a good basis for successfully training an automatic classifier. While pure inter-rater agreement is interesting, we believe that the classification performance measure is a better indicator of quality for training a semantic model. Common agreement measures such as the Kappa statistic may produce results that are biased by trait prevalence and skewedness of the rated categories. This makes them less applicable to datasets such as the one used here, with varying numbers of ratings per image, and a broad range of trait prevalence from approximately 0.1\% for Charts and up to 67\% for Person. The LCM classification performance index $\lambda$, does not suffer from these weaknesses as it is not an agreement measure. In contrast, it is based on the expected classification error that results from the modelling process. We believe that $\lambda$ is therefore better suited for our purposes because it indicates how well we can classify the images based on the given ratings.

The ability to handle varying numbers of ratings per image is an important benefit. As we have shown with the data from the TRECVID 2005 annotation forum, real-world scenarios for image or video annotation cannot always guarantee a fixed number of judgements because of resource constraints. For such cases, the nonresponse model is a robust and flexible solution for evaluating semantic annotations.
Chapter 5

Manual Annotation of Visual Content

“The contestants on one side are those who, briefly stated, believe computers can, should, and will do everything, and on the other side those who, like myself, believe there are limits to what computers ought to be put to do.”

– Computer Power and Human Reason by Joseph Weizenbaum

Manual annotation is used to train supervised learning algorithms for effective video shot classification. In the previous chapter, we proposed to obtain multiple judgements per video shot for modelling a more generic view of the semantics contained in the visual content. This helps to utilise nearly all annotated video shots during training and can possibly lead to more effective decision models for automatic classification. Moreover, the classification performance index that reflects the level of agreement among multiple raters is useful for assessing annotation quality.

Obtaining multiple independent ratings per image increases the labour that is required for the annotation task. Along with the desire to annotate many semantic concepts, we face the organisational challenge to conduct an annotation task efficiently and effectively. We aim to minimise labour, human error, and disagreement between annotators while maximising the number of ratings that we obtain for the most concepts. The advances that supervised learning approaches have provided for multimedia information retrieval have caused a strong need for maximising resource utilisation and annotation accuracy when preparing large training collections. Creating accurately annotated image collections is of high importance to effectively leverage the performance of automatic classifiers.
5.1 THE TRECVID 2005 ANNOTATION EFFORT AT IBM

In this chapter, we focus on efficiency and accuracy aspects when annotating large collections collaboratively. We present outcomes of two user studies that we have conducted to arrive at recommendations to improve vocabulary definition and annotation strategy. In Section 5.1, we outline the organisation and management of the TRECVID 2005 annotation forum at the IBM T. J. Watson Research Center. In particular, we describe the IBM Efficient Video Annotation (EVA) system, its application in the annotation forum, and outcomes of a related user study. A subsequent annotation experiment that we conducted at RMIT University is discussed in Section 5.2. We analyse both annotation experiments with respect to timing and inter-rater agreement for different concepts, and conclude this chapter in Section 5.3 with a summary of the lessons that we have learned.

5.1 The TRECVID 2005 Annotation Effort at IBM

We described the TRECVID 2005 annotation forum in Section 2.3.2 and in Section 4.1. In this section, we focus on the support of the annotation forum at the IBM T. J. Watson Research Center, which included the provision of a new annotation system and conducting a user-study of the annotation effort.

We designed and implemented a new web-based system to overcome difficulties that were identified during and after the 2003 annotation forum [Lin et al., 2003]. These were that annotating on the basis of full video shots was too time consuming, and that allowing a large, uncontrolled concept vocabulary leads to inconsistent, sparse annotations. A team from the Informedia Digital Library Project [Christel et al., 1995] at Carnegie Mellon University (CMU) supported the effort by providing the Informedia Image Classifier. Similar to the system that we developed at IBM, this system implements video annotation on the basis of annotating static images, each representing one video shot. The 137 video clips of the collection to be annotated were therefore temporally segmented into 61,904 distinct shots by a group at the Heinrich-Hertz-Institute (HHI) [Petersohn, 2004], and for each shot, a single representative frame was selected by researchers at Dublin City University [Cooke et al., 2004].

The Informedia Image Classifier is a standalone Microsoft Windows application, while the IBM annotation system is web-based and can be used with standard web browsers. All

\textsuperscript{1}This work was conducted while the author was a visiting researcher at the IBM T. J. Watson Research Center in Hawthorne, NY, USA.

\textsuperscript{2}http://www.informedia.cs.cmu.edu
researchers who volunteered to participate as annotators were given the option to use one or both of the two annotation tools. Over one hundred researchers from 26 institutions volunteered to use the IBM system, allowing for the entire corpus to be annotated with all 39 concepts with the IBM system. Besides leveraging a considerable workforce, this also constituted an organisational challenge. After the participating groups communicated how many person-hours they were able to contribute. The workload was then distributed based on an estimation that one image could be annotated with one concept in approximately one second.\footnote{This was not based on any statistical evidence.} Each of the 26 groups was assigned a subset of video clips and concepts to annotate, so that the allocation of workload is similar to the well-known Knapsack problem [Garey and Johnson, 1979]. In a first step, illustrated in Figure 5.1, we covered the area spanned by the two axes of 137 video clips and 39 semantic concepts. Each square represents a research group consisting of several individual annotators. This step ensures that each key frame is reviewed at least once for each concept. In the second step, illustrated in Figure 5.2, we allocated the remaining groups to have a maximum number of image/concept pairs reviewed again. As a result, we obtained multiple judgements for approximately 44\% of all images. As shown in Figure 5.1, the first allocation step caused some groups to intersect. In combination
5.1. THE TRECVID 2005 ANNOTATION EFFORT AT IBM

<table>
<thead>
<tr>
<th>Category</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. Activities</td>
<td>1. Walking/Running, 2. People Marching</td>
</tr>
<tr>
<td>F. Events</td>
<td>1. Explosion/Fire, 2. Natural Disaster</td>
</tr>
<tr>
<td>G. Graphics</td>
<td>1. Maps, 2. Charts</td>
</tr>
</tbody>
</table>

Table 5.1: All semantic concepts that were used in the TRECVID 2005 annotation effort and their grouping into seven categories. Concepts marked with (*) were dropped because these can either not be annotated reliably with static key frames, or they can be implied from other concepts.

With the second step, this allowed us to collect up to four ratings in some cases. Each group was then given control over how to further sub-divide their overall assignment into individual user assignments so that varying user workloads could be accommodated.

The lexicon of concepts that we used was based on an ontology of semantic concepts that Naphade et al. [2005] defined for TRECVID 2005. In a breadth-first approach, 44 concepts along seven semantic dimensions were selected, with each dimension being as orthogonal as possible to the others. At the same time, the following assumptions were made to promote consistency, simplicity, and speed of annotation:

- Only the terms from the defined concept vocabulary could be used, and no free text annotations were allowed.
- All annotations were for static visual concepts only (as opposed to temporal or aural concepts) and could be inferred from a single key frame without requiring users to play the video clips.
- All annotations were assigned at the global frame level only and were assumed applicable to the entire shot. No object identification or regional annotation was required.

According to the seven semantic dimensions, the concepts are grouped into seven categories, as shown in Table 5.1. These concepts are described in more detail in Appendix B. Concepts
marked with (*) were dropped because these can either not be annotated reliably with static key frames, or they can be implied from other concepts, as is the case for Indoor. 39 out of the 44 defined concepts were finally used in the TRECVID 2005 annotation forum.

Besides the primary goal of creating a large annotated training collection, we wanted to collect statistics such as inter-rater agreement, average annotation time, concept frequency, and progress per concept to study the manual annotation process. These needs, and the aim of minimising the administrative overhead influenced the design of the IBM EVA annotation system.

5.1.1 The IBM EVA Annotation System

Existing video annotation applications, such as the VideoAnnEx system [Lin et al., 2003] that was used for the first annotation effort in 2003, did not seem suitable for the TRECVID annotation forum. Our goal was to simplify and speed up the annotation process, while maintaining sufficient configuration and customisation options to allow for different annotation styles and user preferences, which is required for a large number of annotators. Moreover, we wanted to minimise the administrative overhead, and ensure high annotation quality. We therefore developed the IBM Efficient Video Annotation (EVA) system\textsuperscript{4} for the 2005 TRECVID annotation forum. This system has also been used in the LSCOM annotation effort [Kennedy et al., 2006; Naphade et al., 2006] to annotate the TRECVID 2005 development corpus with up to 1000 concepts. The EVA system is designed to encourage annotation with only one or a few concepts at a time for a given set of images or frames, thus promoting completeness in the annotation process.

Different users may be most comfortable and most efficient using different modes of annotation. For example, annotating only a few images per page without the need to scroll versus annotating many images at a time and scrolling as required. Moreover, while some users might prefer exhaustively annotating with a single concept before proceeding to the next concept, others may wish to annotate with several concepts simultaneously. To maximise user convenience and efficiency, each user has the option to customise the number, size, and layout of thumbnails displayed per page. This can be done in the main menu, as shown in Figure 5.3, before an annotation session commences.

The annotation progress is displayed for the current concept and set of video shots during annotation. In addition, the overall progress is shown in the main menu, and detailed

\textsuperscript{4}http://domino.watson.ibm.com/comm/research.nsf/pages/r.multimedia.innovation.html
5.1. THE TRECVID 2005 ANNOTATION EFFORT AT IBM

Figure 5.3: The start menu of the EVA system.

Figure 5.4: Users can view individual annotation statistics.

statistics can be generated, as shown in Figure 5.4. These show the statistics of the current annotation status for each video and each concept assigned to the current user. An administrator may view these progress statics for each user account.

The annotation process can be controlled entirely by mouse or by keyboard. The use of a keyboard is still rare among web-based applications, but preliminary experiments among several TRECVID participants have indicated that the annotation can be performed more efficiently by using a keyboard once a user has undergone brief training. Each user also has the choice between one or multiple concepts to annotate at a time. Figure 5.5 shows a screenshot of the EVA system during an annotation session. As illustrated in this figure, we aimed at utilising the screen real estate optimally so that the focus is on the content being annotated.

Considering lessons learned from the 2003 annotation forum, we also targeted higher annotation quality by using a server-based architecture, allowing the collection of background statistics on the collaborative annotation effort. These statistics include the average time that a user requires to annotate an image, and the completion status of the annotation task for each user. Other important requirements were the assignment of specific workloads to individual users and the ability to shift work between users should this become necessary.

The concept lexicon can be loaded on the server side and each annotator can be assigned either the full lexicon or a part thereof. Similarly, particular video clips can be allocated to each user so that we can define annotation tasks for individual annotators. A user then has access to only the videos and concepts assigned to them. An annotation session is
Figure 5.5: The main annotation screen of the EVA system. Each video shot is represented by a static key frame. The frames are organised on several pages through which users can navigate. Each key frame may be annotated as either “positive”, “negative”, “ignore”, or “skip” in regards to the currently selected concept that is currently selected at the upper left of the screen.

Comprised of annotating one video clip with one or more concepts out of those available. During a session, annotators can navigate page by page through the entire set of key frames. By default, these are presented in temporal order to maintain context within a video clip, although the system does allow randomisation.

As described in Section 4.1, each image can be annotated as “positive”, “negative”, or “skip” in regards to the currently selected concept. The label “skip” is the default state and means that the image should neither be used as a positive nor as a negative training example. However, this does not allow to distinguish between images that have explicitly been set to “skip” and those that have not been reviewed. The EVA system therefore allows users to assign a fourth label “ignore” to indicate that the image has been reviewed but should not be used as a training example for the given concept. After the annotation effort, we combined “ignore” and “skip” annotations into one group. Ideally, there should not have
been any “skip” annotations but not all images could be reviewed due to time constraints in some participating groups. The final group “skip” thus contains video shots that should not be used in training because they were explicitly excluded and video shots that should not be used because they were not reviewed. As can be inferred from the information in Table 4.2 in the previous chapter, “skip” annotations formed on average 1.7% of the collection across all concepts.

The assignment of labels is done for one concept at a time; the user decides which and how many concepts are used in the current session. We believe that this leads to a more accurate and complete annotation as opposed to annotating all available concepts at once. Past experience has shown that the latter can cause many concepts to be missed. Bulk annotation buttons are available for all of the four labelling states. They allow annotators to label shots of an entire page with either “positive”, “negative”, “ignore”, or “skip” in regards to the current concept. These buttons act only on previously unlabelled thumbnails, except for the bulk “skip” button which clears all annotations for a given concept on that page. We believe that bulk labelling enables more efficient annotation for very rare or very frequent concepts by assuming a default state of “negative” or “positive” labels, and correcting only the few examples that are incorrect. Very rare concepts may still be more difficult to annotate reliably and thoroughly since they require more careful annotation in order to avoid missing relevant images.

When using the keyboard for annotation, a cursor is used to navigate between thumbnails; labels can be assigned by using only a single keystroke. If a label is assigned by keyboard, the cursor is automatically advanced to the next thumbnail. As illustrated in Figure 5.6, a high resolution version of each image may be displayed in an overlay window to allow better determination of details.

In accordance with the simplifications for the 2005 annotation effort that we described on page 102, the EVA system currently supports only global annotation, with each label assigned to the entire image rather than to a part of the image. The labelling of events in the temporal space is currently not supported as we use only still images. However, some context is provided through neighbouring thumbnails unless the presentation is randomised. To facilitate a user study, the EVA system has functionality to collect aggregate user data during the annotation. This data includes the time spent on each page, the number and size of thumbnails, and statistics about the usage of keyboard and mouse. The fact that we have access to all annotation data on the server complements this feature well, and allows us to compile valuable statistics during and after annotation.
Figure 5.6: Each frame can be viewed enlarged in an overlay window to identify details.

5.1.2 Data Evaluation and Results

To ensure maximum quality of the annotations, and to provide optimal guidance for annotators, we must be able to assess the complexity, or the difficulty, of the semantic concepts that we use. The ability to identify problematic concepts helps to optimise the definition of semantic concepts for the future, and contributes to more accurately annotated training collections. We have already reduced the complexity of the task by not annotating events that expand into the temporal space. However, concepts, such as People-Marching or Walking/Running are events with a temporal dimension and were part of the TRECVID 2005 concept vocabulary. These concepts were included because participants agreed that static key frames provide enough context to annotate these concepts without reviewing the full shot. We believe that such concepts are more complex to annotate than clearly defined objects, such as Car or Airplane. A second group of concepts that, we believe, are difficult to annotate are concepts that describe a role, such as Corporate Leader, Government Leader, and Police/Security. These concepts usually require more thorough examination of the image.
by the reviewer to identify the person and their role. Moreover, domain knowledge of politics, current affairs, and business is often required to identify the role of a particular person. The prevalence of a concept may also have an influence on difficulty of annotation. For example, very infrequent concepts may be more difficult to annotate because a training effect is less likely to occur, and annotators may find it harder to spot positive examples.

We assess concept difficulty by evaluating the average annotation time and the average inter-rater agreement per concept. While annotation times above average are normally not problematic from a quality standpoint, they may indicate that users had difficulty annotating this concept and we want to be able to identify the reasons for this. Some concepts require detailed inspection of images, which can cause users to take longer to annotate these concepts. It is also possible that concept definitions are not clear enough in some cases, or that a concept is perceived differently by individual annotators due to regional or cultural differences. In this case, we want to be able to improve concept definitions and provide better guidelines for annotators in the future to avoid inconsistencies. Concept ambiguity may not necessarily cause annotators to require more time for the annotation, but it is likely to have an impact on the annotation quality. That is, we are likely to observe higher disagreement in the user ratings if the interpretation of a concept differs significantly between the raters. This can particularly be an issue in global annotation efforts such as the TRECVID annotation forum.

As discussed in the previous chapter, traditional inter-rater agreement measures may not always yield reliable results for our data sets. The classification performance index $\lambda$ that is computed as part of the latent class modelling process is more suitable for assessing annotation quality. We will therefore use the classification performance index $\lambda$ instead of the inter-rater agreement. We expect users to require more time to annotate difficult concepts, and poorly specified concepts to be accompanied by low classification performance that is caused by disagreement between ratings. We plot the classification performance index $\lambda$ over the average annotation time per image for all concepts; this is shown in Figure 5.7. In this graph, each of the seven concept categories is represented by a symbol as shown in the legend. The category averages are shown as large symbols, while the individual concepts appear as smaller symbols. Individual concepts are labelled with the letter for the category that they belong to and their individual index, as shown in Table 5.1. We calculate the average annotation times $\bar{t}$ and average classification performance $\bar{\lambda}$ for all concepts, including the standard deviations $\sigma_t$ and $\sigma_\lambda$. We then define a maximum threshold for the annotation time $t_{max} = \bar{t} + \sigma_t$, and a minimum classification performance threshold $\lambda_{min} = \bar{\lambda} - \sigma_\lambda$. Based on these thresholds, we identify concepts for which the average annotation time is greater
Figure 5.7: The average classification performance index $\lambda$ over average annotation time per concept. We defined a standard area (shaded grey) based on the overall averages and their standard deviations. Concepts outside this area are regarded as outliers; these are more interesting for our discussion.

than $t_{max}$ or the classification performance is below $\lambda_{min}$. We computed the thresholds to be $t_{max} = 2.95s$ and $\lambda_{min} = 0.61$.

The majority of concepts is found within the ranges defined by our thresholds; these appear in the shaded area in Figure 5.7. We do not regard concepts as outliers if they were annotated in relatively short time. This is the case for all concepts of category A (program category): Sports (A4), Entertainment (A5), and Weather (A6); and for some concepts of category B (setting/scene/site): Mountain (B10), Road (B11), Snow (B13), and Urban Setting (B14). Category A can be annotated in short time because these usually appear temporally clustered, and the EVA annotation system preserves temporal continuity. Sections of a particular program category can thus be identified efficiently. The four concepts from category B that were annotated relatively quickly are visually distinctive and can therefore be identified well without more detailed inspection of a frame. Moreover, the specifications of these concepts seem to be generally very clear and we can regard these concepts as not
5.1. THE TRECVID 2005 ANNOTATION EFFORT AT IBM

problematic. This is supported by the fact that we observe relatively high classification performance for all of these concepts.

The concepts that are found outside the area marked by $t_{\text{max}}$ and $\lambda_{\text{min}}$ — the white area in Figure 5.7 — deserve more attention because these concepts seem to be more difficult or problematic. The concepts Vegetation (B9), Desert (B8), Police/Security (C6), and Corporate Leader (C5) are problematic because we observe particularly low classification performance for these. We expected that Police/Security and Corporate Leader would be annotated with low agreement, and thus with low classification performance. As we have discussed above, these concepts require detailed inspection of frames and deeper domain knowledge, often concerning a particular region. This domain knowledge is most likely inconsistent within a multinational and multi-cultural group of annotators such as the one participating in the TRECVID 2005 annotation forum. Interestingly, we observe relatively high classification performance for the concept Government Leader (C4) that we expected to be similarly difficult to annotate. This can be explained by the TRECVID collection featuring mostly news footage covering world politics. The knowledge of high-ranking government officials is more likely to be consistent among the group of annotators. The classification performance for the concepts Vegetation and Desert is not as poor as for Corporate Leader or Police/Security, but still well below $\lambda_{\text{min}}$. These concepts may imply more vagueness than we initially expected.

Annotators may have differed on the relevance of many images because these concepts appear often in the background. But this is also the case for concepts such as Road (B11), Sky (B12), and most other concepts of category B. We cannot draw reliable conclusions from the evidence at hand, and further studies are necessary. Similarly, the concept Office (B3) of category B has low classification performance and required a relatively long time to be annotated. Again, a reason for this might be different interpretations of what an office setting should look like, but our data does at this point not permit a reliable explanation.

Two of the clearly specified objects, Truck (D7) and Boat/Ship (D8), appear to the right of the grey shaded area of the graph in Figure 5.7. This means that these concepts took relatively long to annotate. For the concept Truck, this is accompanied by low classification performance. The lack of specificity in the definition of Truck is a likely cause for the low classification performance. In the English language, a Truck can either be a large freight vehicle or a small car-like vehicle with an open load floor. In contrast, the definition of Boat/Ship is very clear and the classification performance for this concept is high. The reason for the increased annotation times for the concepts Boat/Ship and Truck can be found when considering the number of concepts that users simultaneously annotated. As we show below,
we identify a dependency between the annotation time and the number of concepts that users annotated: the more concepts users choose to annotate simultaneously, the higher the likelihood that these require more time. We calculate the average number of concepts that users simultaneously annotated. Among all users, this average is 3.7 concepts. For users who annotated the concept Boat/Ship, the average is 6.3 concepts, and the average among users who annotated Truck is 6.3 concepts. While this explains the higher annotation times for these two concepts, it is difficult to explain why the annotation times for the concepts Outdoor (B6), Court (B2), and Office were even higher. For these, the average numbers of simultaneously annotated concepts are 3.2 concepts, 3.4 concepts, and 2.9 concepts. This is well below the overall average and there are most likely other factors that we cannot identify with the data at hand. We excluded the varying concept prevalences as a significant factor when we plotted the average annotation time as a function of concept prevalence. We did not observe any clear pattern or trend, and do therefore not show this graph here.

We plot the average annotation time over the average number of selected concepts on a per-user basis in Figure 5.8 to illustrate the effect of the number of simultaneously selected concepts on the annotation time. Each data point represents a user’s average annotation time $t$ per image/concept pair. We compute the average $\bar{t}$ and the standard deviation $\sigma_t$ over all users, and define a threshold at $t_{max} = \bar{t} + \sigma_t$ to identify outliers. Here, we compute $t_{max} = 4.00s$; the area below this value is shaded grey in the graph in Figure 5.8. As can be seen from this figure, many users chose to annotate with a single concept which results in a cluster of data points in the lower part of the y-axis. We generally observe substantial variation in annotation times per image. However, the number of outliers increases when users choose to annotate with more concepts simultaneously. The difference of the annotation times of the outliers to the average tends to increase in tandem with this. To better illustrate this effect, we compute a linear trend through all data points, as shown in Figure 5.8. This trend indicates a moderate correlation between the number of simultaneously annotated concepts and the average annotation time; the Pearson’s product-moment correlation coefficient is $\rho = 0.51$.

However, it is still difficult to draw strong conclusions based on our data because the experiment was not conducted in a controlled environment; we cannot confirm that users who selected many concepts really annotated these simultaneously. The EVA annotation system allows a user to revert to annotating only one concept at a time despite selecting many concepts. Moreover, the choice of the input device by a user, and the number of thumbnails per page will also have influenced the annotation time.
5.1. THE TRECVID 2005 ANNOTATION EFFORT AT IBM

![Figure 5.8](image)

Figure 5.8: The average annotation time per image over the number of concepts that users annotated simultaneously. The area shaded in grey marks the overall average plus the standard deviation. Data points outside this area are regarded as outliers. The linear trend shows that the annotation time tends to increase with the number of simultaneously annotated concepts.

We illustrate the influence of the input device on the annotation time in Figure 5.9. We recorded a count of the mouse clicks and a count of the keystrokes for each user per page during the annotation. We group users based on the ratio of mouse clicks to keystrokes. Users with a ratio under 0.5, are in group “Keyboard”. This means that these users have used the keyboard at least twice as often as the mouse. A ratio of 2.0 and above means that these users have used the mouse at least twice as often as the keyboard. Accordingly, these users are in group “Mouse”. The users with a ratio between 0.5 and 2.0 are in the group “Both” as their preferred input device could not clearly be classified. From Figure 5.9, it can be seen that users who mostly used the keyboard required less time compared to users who mostly used the mouse or a combination of mouse and keyboard. Still, the largest group of users (47%) preferred the mouse as input device. The keyboard was preferred only by 19% of all users, while 22% used both.
Finally, we evaluate the average annotation time as a function of the number of thumbnails per page that users chose to display while annotating. Figure 5.10 shows the graph that resulted from this evaluation. In this graph, a diamond symbol represents a user’s average annotation time, while a square symbol represents the number of thumbnails that the user chose to be displayed per page. We order the users from left to right by the number of thumbnails per page in ascending order. As Figure 5.10 shows, there is little dependency between the annotation speed and the number of thumbnails per page. Assuming a linear trend for the average annotation time, as shown in Figure 5.10, suggests a weak tendency in favour of using more thumbnails per page. However, there is no statistically significant correlation: the correlation factor between the number of thumbnails per page and the average annotation time is $\rho = -0.07$.

5.1.3 Considerations on the TRECVID Results

Many factors influence our results on the TRECVID 2005 annotation data. It is thus difficult to draw reliable conclusions, but we observe interesting trends. We believe that the assessment of inter-rater agreement and annotation time on a per-concept basis allows inference of concept difficulty. While we observe some interesting outliers, concepts that concern events, activities, or roles can be identified as difficult to annotate. Even outliers such as Airplane, Truck, and Bus can be explained when taking into account that some users annotated with
more concepts than others. Some concepts appear to be specified too vaguely, as indicated by low inter-rater agreement in the annotations of such concepts.

Our results do not suggest that there is a dependency between concept prevalence and annotation time. We could confirm that on average, users who prefer to use the keyboard annotate faster than users who use the mouse, and that the number of thumbnails that are displayed per page has no influence on the average annotation time.

We deduce from Figure 5.8 that users may require more time to annotate if they choose very many concepts. However, the evidence for this is relatively weak because of the many different factors that may have influenced our results. Another important question is whether the number of concepts that users simultaneously annotate has an influence on the inter-rater agreement. Confirmation of such a dependency would help to maximise inter-rater agreement in collaborative annotation efforts. Since the TRECVID 2005 annotation data does not allow reliable conclusions about this or most of the issues described above, we conducted another annotation experiment. We now describe this subsequent experiment and its outcomes.
5.2 Annotation Experiments at RMIT University

We designed a second experiment to study human judgement of images in a controlled environment. In contrast to the TRECVID 2005 annotation effort, we did not aim to generate an annotated collection of examples for training. The primary goal of our experiment was to study the effects on efficiency and inter-rater agreement of the following two factors:

- Concept vocabulary: Specific objects, such as Car or Airplane, that are less prevalent versus different shot settings with higher prevalence, for example Vegetation or Sky.

- Annotation mode: Annotating one image in regards to a single concept at a time versus annotating one image with five concepts simultaneously.

Figure 5.11 shows a screenshot of the web-based Sapphire annotation system, specifically designed for this task. Sapphire supports annotation of one still image at a time with either one or several concepts. To relate our conclusions back to our earlier work at TRECVID, we selected images and semantic concepts for the annotation from the collection that was used in the TRECVID 2005 annotation forum. That is, we randomly chose a sample of 600
5.2. ANNOTATION EXPERIMENTS AT RMIT UNIVERSITY

<table>
<thead>
<tr>
<th>Concept set $a$</th>
<th>Concept set $b$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>vehicles</strong></td>
<td><strong>settings/scenes/sites</strong></td>
</tr>
<tr>
<td>$a_1$ Car</td>
<td>$b_1$ Outdoor</td>
</tr>
<tr>
<td>6%</td>
<td>33%</td>
</tr>
<tr>
<td>$a_2$ Truck</td>
<td>$b_2$ Sky</td>
</tr>
<tr>
<td>2%</td>
<td>13%</td>
</tr>
<tr>
<td>$a_3$ Bus</td>
<td>$b_3$ Studio</td>
</tr>
<tr>
<td>$&lt;1%$</td>
<td>11%</td>
</tr>
<tr>
<td>$a_4$ Airplane</td>
<td>$b_4$ Building</td>
</tr>
<tr>
<td>1%</td>
<td>12%</td>
</tr>
<tr>
<td>$a_5$ Boat/Ship</td>
<td>$b_5$ Vegetation</td>
</tr>
<tr>
<td>$&lt;1%$</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 5.2: The two sets of concepts, $a = \{a_1, \ldots, a_5\}$ and $b = \{b_1, \ldots, b_5\}$, that we have used in our experiment along with the expected prevalences based on TRECVID experiments.

images and selected ten of the 39 semantic concepts such that we obtain two groups: specific objects with low prevalence and more vaguely specified settings, sites, or scenes with higher prevalence. We estimated the concept prevalences based on the data collected in 2005; the semantic concepts and the expected prevalences in our randomly selected collection sample are shown in Table 5.2. The semantic concept definitions are described in more detail in Appendix B and the set of images that we used is described below.

5.2.1 Objectives and Methodology

We invited 20 users, 10 male and 10 female, to participate as annotators in our experiment. The participants were mostly drawn from the general public. Five of our participants were research students of the Science, Engineering, and Technology Portfolio at RMIT University. All users were familiar with the use of computers, but mostly not experienced in the field of multimedia information retrieval. The experiment was conducted anonymously, with each user randomly assigned an anonymous user account such that responses could not be traced back to individual participants.

We allowed a brief training phase so that the participants could familiarise themselves with the system before the experiment. All users were presented one image at a time and (depending on the annotation mode) either one concept or five concepts next to the image. The task required users to select all the concepts that they considered to be depicted in the image. The annotation system offers navigation buttons for users to go backwards and forwards between images. However, we divided the image collection into several sets as described below, and free navigation forward and backwards was only allowed within one image set. We introduced this restriction to provide a minimum of guidance through the
collection while allowing users to correct any errors they might make. Each user performed the experiment in two parts; one where they annotated multiple sets of images, each with a single concept that is varied between sets, and one where they annotated one set of images with multiple concepts. For a given user, neither the images nor the concepts overlapped between parts. During the experiment, we recorded the annotation generated by each user in a central database. Additionally, we measured the time that each user spent per image. After the experiment we asked users to complete a short survey with simple questions about the annotation.

We randomly selected 600 images out of the TRECVID 2005 development collection [Over et al., 2005] for our experiment. We divided the image collection into two equal parts, $A$ and $B$, each comprising 300 images. Each set was then further divided into five subsets of 60 images each $A = \{A_1, \ldots, A_5\}$, and $B = \{B_1, \ldots, B_5\}$. For the ten selected concepts, we defined two sets $a$ and $b$. These were selected such that one set $a = \{a_1, \ldots, a_5\}$ represents well-known objects that appear relatively rarely in the collection, while the second set $b = \{b_1, \ldots, b_5\}$ consisted of settings and scenes with a significantly higher prevalence, as can be seen from Table 5.2.

Our experiment setup is illustrated in Figure 5.12. We grouped the 20 participants into four groups of five users each, and paid attention to achieving a uniform distribution of male and female participants among all groups. In Figure 5.12, a concept/image set combination such as $a_1A_3$ means that the user annotates the image set $A_3$ with concept $a_1$ (Car). Similarly, $b_1b_2b_3b_4b_5B_3$ means that the user annotates image set $B_3$ with all five concepts of concept group $b$. Each user therefore annotated six subsets of images, that is 360 images: five subsets with a single concept (single-concept mode) each, and one subset with five concepts simultaneously (multiple-concepts mode). The experiment was divided into two parts so that each user was to use both annotation modes. To cancel out unwanted effects, we rotated the order of annotation modes among user groups. We also rotated the order in which users would see images to avoid training effects influencing our results.

This setup enables us to evaluate concept prevalences and inter-rater agreement based on the different annotation modes as well as in combination. Moreover, we can draw conclusions about which mode might be faster, and can make a more reliable statement about the impact of different concept types on the annotation.
5.2. ANNOTATION EXPERIMENTS AT RMIT UNIVERSITY

<table>
<thead>
<tr>
<th>Group</th>
<th>User</th>
<th>Part 1</th>
<th>Part 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>user1</td>
<td>a1A1, a2A2, a3A3, a4A4, a5A5</td>
<td>b1 b2 b3 b4 b5 B1</td>
</tr>
<tr>
<td></td>
<td>user2</td>
<td>a1A2, a2A3, a3A4, a4A5, a5A1</td>
<td>b1 b2 b3 b4 b5 B2</td>
</tr>
<tr>
<td></td>
<td>user3</td>
<td>a1A3, a2A4, a3A5, a4A1, a5A2</td>
<td>b1 b2 b3 b4 b5 B3</td>
</tr>
<tr>
<td></td>
<td>user4</td>
<td>a1A4, a2A5, a3A1, a4A2, a5A3</td>
<td>b1 b2 b3 b4 b5 B4</td>
</tr>
<tr>
<td></td>
<td>user5</td>
<td>a1A5, a2A1, a3A2, a4A3, a5A4</td>
<td>b1 b2 b3 b4 b5 B5</td>
</tr>
<tr>
<td>2</td>
<td>user6</td>
<td>b1 b2 b3 b4 b5 B1</td>
<td>a1A1, a2A2, a3A3, a4A4, a5A5</td>
</tr>
<tr>
<td></td>
<td>user7</td>
<td>b1 b2 b3 b4 b5 B2</td>
<td>a1A2, a2A3, a3A4, a4A5, a5A1</td>
</tr>
<tr>
<td></td>
<td>user8</td>
<td>b1 b2 b3 b4 b5 B3</td>
<td>a1A3, a2A4, a3A5, a4A1, a5A2</td>
</tr>
<tr>
<td></td>
<td>user9</td>
<td>b1 b2 b3 b4 b5 B4</td>
<td>a1A4, a2A5, a3A1, a4A2, a5A3</td>
</tr>
<tr>
<td></td>
<td>user10</td>
<td>b1 b2 b3 b4 b5 B5</td>
<td>a1A5, a2A1, a3A2, a4A3, a5A4</td>
</tr>
<tr>
<td>3</td>
<td>user11</td>
<td>a1 a2 a3 a4 a5 A1</td>
<td>b1B1, b2B2, b3B3, b4B4, b5B5</td>
</tr>
<tr>
<td></td>
<td>user12</td>
<td>a1 a2 a3 a4 a5 A2</td>
<td>b1B2, b2B3, b3B4, b4B5, b5B1</td>
</tr>
<tr>
<td></td>
<td>user13</td>
<td>a1 a2 a3 a4 a5 A3</td>
<td>b1B3, b2B4, b3B5, b4B1, b5B2</td>
</tr>
<tr>
<td></td>
<td>user14</td>
<td>a1 a2 a3 a4 a5 A4</td>
<td>b1B4, b2B5, b3B1, b4B2, b5B3</td>
</tr>
<tr>
<td></td>
<td>user15</td>
<td>a1 a2 a3 a4 a5 A5</td>
<td>b1B5, b2B1, b3B2, b4B3, b5B4</td>
</tr>
<tr>
<td>4</td>
<td>user16</td>
<td>b1B1, b2B2, b3B3, b4B4, b5B5</td>
<td>a1 a2 a3 a4 a5 A1</td>
</tr>
<tr>
<td></td>
<td>user17</td>
<td>b1B2, b2B3, b3B4, b4B5, b5B1</td>
<td>a1 a2 a3 a4 a5 A2</td>
</tr>
<tr>
<td></td>
<td>user18</td>
<td>b1B3, b2B4, b3B5, b4B1, b5B2</td>
<td>a1 a2 a3 a4 a5 A3</td>
</tr>
<tr>
<td></td>
<td>user19</td>
<td>b1B4, b2B5, b3B1, b4B2, b5B3</td>
<td>a1 a2 a3 a4 a5 A4</td>
</tr>
<tr>
<td></td>
<td>user20</td>
<td>b1B5, b2B1, b3B2, b4B3, b5B4</td>
<td>a1 a2 a3 a4 a5 A5</td>
</tr>
</tbody>
</table>

Figure 5.12: The specification of our experiment, each user annotated 360 images. For each image/concept pair, we obtain four ratings. Lowercase letters represent concepts, while uppercase letters each represent a set of 60 images.

5.2.2 Evaluation and Results

The experiment setup results in four independent ratings per image, two for each of the single-concept and multiple-concepts annotation modes. Due to the total collection size of 600 images, we can evaluate 300 distinct images for each of the ten concepts based on four ratings. All concept prevalences reported here are therefore relative to a collection size of 300 images.

Based on our prior experience, we anticipated the annotation speed to be faster if users had to annotate only one concept as opposed to five concepts at a time. We also expected the agreement between raters to be higher when annotating only a single concept. In addition to these effects, we expected users to have greater difficulty annotating the second set of concepts, set b, because these leave more room for ambiguity. We thus expected a higher disagreement rate for these concepts.

When evaluating annotations, we are primarily interested in the positive ratings because these determine which images may be used as positive examples for training. All other images are usually considered negative examples. In our case, each image could be assigned
Figure 5.13: The raw positive ratings for each concept, ordered by the concept prevalence of each concept as estimated with our latent class model. Each bar represents the number of images that received 1, 2, 3, and 4 positive ratings for the respective concept. We observe substantial disagreement in the ratings.

a maximum of four positive ratings. The raw positive ratings that we obtained during our experiment are illustrated in Figure 5.13. This graph again highlights the difficulty when classifying visual content. As can be seen, we observe substantial disagreement between the ratings for all concepts.

We apply latent class modelling, as described in Section 4.2.1, to assess the concept prevalences and classification performance. As in the previous chapter, we compare the LCM-based classification performance to the inter-rater agreement based on Fleiss’ Kappa measure because this agreement measure is well known and widely used for such purposes. The only difference to the evaluation described in the previous chapter is that we now have a fixed number of four observed ratings per image. We can thus apply the generic latent class model as defined in Equation 4.3 and are not required to handle nonresponse. However, we still assume interchangeability of raters because the four ratings for a given image do not always originate from the same four annotators.
Table 5.3: Concept prevalences computed with latent class modelling, using modal classification, for our annotation experiment. The percentages are reported for the sample size of 300 images. The results correlate with the predicted concept prevalences that were based on the TRECVID 2005 annotation data, except for Studio.

Inter-rater Agreement and Classification Performance

We compare the LCM-based classification performance $\lambda$ to the inter-rater agreement in Figure 5.15. As in the previous graphs, we order the concepts by their frequency in the collection with the most frequent concept on the right. We observe generally significant correlation between the classification performance index $\lambda$ and the inter-rater agreement $\kappa$, but with substantial differences for the least frequent concepts Bus and Boat/Ship. Interestingly, $\lambda$ indicates perfect classification for the concepts Bus and Boat/Ship, while $\kappa$ suggests only average to poor agreement. As we discussed in Section 4.2.2, the Kappa inter-rater agreement statistic is known to be biased by trait prevalence and may yield unreliable results when only few positive examples are found in the collection. To support our hypothesis that Kappa is
Figure 5.14: The prevalences of all concepts in our annotation experiment, based on latent class modelling with four ratings per image. The light blue bars represent the estimates by the latent class model, the dark blue bars represent the number of positive examples obtained using modal classification.

not suitable for assessing annotation quality for rare concepts in our collection, we refer to the raw ratings, as shown in Figure 5.13. For the concepts Bus and Boat/Ship, we observe response patterns that by intuition appear rather unambiguous and allow good classification. For Boat/Ship, all four raters agree on “positive” in two cases, and in six cases three raters vote “negative” against a single “positive” rating. Almost identically, for the concept Bus all four raters agree on “positive” in one case, while three raters vote “negative” against a single “positive” rating in six cases. The latent class model can resolve the ambiguity very reliably, and the estimated classification performance is $\lambda = 1.0$ despite the fact that there is some disagreement. We believe that the results for concepts Bus and Boat/Ship in terms of $\kappa$ are heavily biased by the low trait prevalence of these concepts, and that the results are therefore not reliable.

In contrast, $\kappa$ and $\lambda$ are consistently low for the concepts Studio and Vegetation. We described above that the concept Studio may have been difficult to annotate for the participants.
of our experiment because we do not expect the general public to have a clear imagination of such a concept. Indeed, in the survey that we conducted in connection with our experiments, 50% of all participants stated that they found it difficult or very difficult to annotate the concept Studio. Figure 5.16 shows the summarised results of this survey, in which we have asked all participants to rate the concept difficulty using the categories very easy, easy, difficult, and very difficult. From this figure, it can be seen that the concept Studio was rated the most difficult concept. Remarkably, only 15% of all users rated Vegetation as difficult to annotate. However, the concept Vegetation was consistently annotated with low inter-rater agreement in both the TRECVID annotation effort and our experiment at RMIT University. It appears that individual users have no difficulty to decide whether an image is a positive or a negative example for Vegetation, but that the term “Vegetation” by itself leaves too much room for interpretation. The concept definition requires additional clarification in such a case.

In the next step, we analyse the inter-rater agreement separated by annotation mode to test whether there are any differences in the annotation quality between both modes. For
Figure 5.16: The concept difficulty as rated by the participants of our annotation experiment. Each annotator was asked to rate the difficulty of each concept using the categories very easy, easy, difficult, and very difficult.

Each image, we obtained four judgements, two while the users annotated a single concept for each image, and two while multiple concepts were annotated. We hypothesised that annotating with many concepts simultaneously might increase the error rate, and that this could be reflected by lower inter-rater agreement in multiple-concepts mode. We revert to using only the $\kappa$-based inter-rater agreement for this analysis because we cannot estimate a two-class latent class model such as ours with only two manifest variables. Such a system has zero degrees of freedom and any result would be inconclusive. Hence, we cannot compute $\lambda$ to conduct this analysis, and so we use the Kappa statistic instead [Fleiss, 1981].

The results for $\kappa$ in single-concept mode (yellow bars) and multiple-concepts mode (blue bars) are shown in Figure 5.17. Again, the evaluation of the inter-rater agreement for the very rare concepts is problematic. The situation in this analysis has worsened because we observe even fewer positive ratings when evaluating annotations separately for each annotation mode. This causes the results for the three least prevalent concepts Bus, Boat/Ship, and Airplane most likely to be unreliable. For example, the value of $\kappa = -0.005$ for the concept Airplane in
5.2. ANNOTATION EXPERIMENTS AT RMIT UNIVERSITY

![Figure 5.17: Inter-rater agreement ($\kappa$) for annotations done in single-concept mode compared to the intraclass correlation for annotations in multiple concept mode. We observe no statistically significant differences between both when evaluating the results for the seven most frequent concepts. We ignore the three most infrequent concepts due to possibly invalid results for $\kappa$.](image)

single-concept mode results from three positive ratings that are all observed in disagreement. In this case, the Kappa statistic reports agreement below chance, that is, the agreement is lower than the agreement that can be expected by random assignment. However, this neglects the fact that all other ratings agreed on the remaining 297 images being negative examples. We therefore treat the concepts *Bus*, *Airplane*, and *Boat/Ship* as outliers in this analysis and indicate this by showing these concepts in light grey in Figure 5.17.

When evaluating the results for $\kappa$ of the remaining seven concepts, we observe differences between single-concept mode and multiple-concepts mode. The annotations that were done in single-concept mode have a higher inter-rater agreement rate in most cases. While this seems to support our hypothesis that annotating several concepts at a time leads to higher disagreement, we could not confirm statistical significance of the differences using the two-tailed Student T-Test. We can therefore not confirm this hypothesis based on our results.
Figure 5.18: The annotation time per image by each user in absolute terms. Annotating only one concept per image faster than annotating five concepts per image. Interestingly, users 1 to 10 annotated their concepts in multiple-concepts mode much slower than users 11 to 20. This is because users 1 to 10 annotated the more frequent concepts of group $b$ in multiple-concepts mode.

However, this conclusion is on the basis of using five concepts simultaneously and it is likely that annotators become overburdened when the number of concepts is increased. Based on our current observations, we conclude that annotating five concepts at a time is well within the capacity of the average human annotator. Further studies are needed to determine a maximum number of simultaneous concepts that can be annotated while maintaining acceptable quality.

Similarly, we cannot confirm a dependency of the annotation quality on the concept vocabulary based on our data. For this evaluation, we again use all four ratings and compare the classification performance index $\lambda$ of the two concept groups $a$ and $b$ that we used in our experiment. Concept group $a$ includes objects — in this case vehicles — that can be described as clearly specified, while concept group $b$ included different settings and scenes that leave more room for individual interpretation. The average classification performance differs between both groups: the average classification performance for concept group $a$
5.2. ANNOTATION EXPERIMENTS AT RMIT UNIVERSITY

Figure 5.19: The Annotation time per image and per concept by each user. Annotating the more frequently occurring concepts took longer on average than annotating the less frequent ones. Overall, annotating one concept per image is done quickest when multiple concepts are annotated simultaneously.

is $\lambda_a = 0.9547$, while the classification performance for concept group $b$ is $\lambda_b = 0.8845$. However, using the two-tailed Student T-Test, we cannot establish statistical significance of the classification performance results between the two groups. While this answers the question about whether there is a dependency of the annotation quality on the concept vocabulary specifically for our experiment, further experiments are needed to find a more general answer.

**Annotation Efficiency**

Timing data was recorded during the annotation experiment. We measured the times that users spent while annotating each image. The average times are illustrated in Figure 5.18. We separate users 1 to 10 and users 11 to 20 into different groups because users 1 to 10 annotated the more prevalent concept group $b$ in multiple-concepts mode, while users 11 to 20 annotated the significantly less frequent concepts of group $a$ in multiple-concepts mode. We indicate the average times for both single and multiple-concepts modes in Figure 5.18. On average,
users 1 to 10 spent 5.43 seconds per image annotating five concepts while users 11 to 20 spent 2.17 seconds on average per image annotating five concepts. This shows that the users required on average longer to annotate the more frequent concepts. For the same reason, users 1 to 10 were quicker in annotating in single-concept mode as they annotated the less prevalent concepts in this mode. Users 11 to 20 needed 1.75 seconds per image on average to annotate the more frequent concepts, while users 1 to 10 needed on average 1.19 seconds for the less frequent concepts.

This is supported when evaluating the relative annotation times. We calculate the average annotation time needed per image and per concept in both modes and compare them in Figure 5.19. Again, we indicate the averages for users 1 to 10 and users 11 to 20 in both modes. We observe the shortest times for users 11 to 20 when they annotated the rare concept set a in multiple-concepts mode; only 0.48 seconds were needed on average per image and concept. Users 1 to 10 needed 1.09 seconds on average to annotate the more prevalent concepts of group b in combination. In conclusion, we can report that it is more efficient to annotate in multiple-concepts mode. Given that navigating between images incurs an overhead, this is not surprising. Specialised implementations may address this by automatically guiding annotators to the next image after annotating one concept, but we do not believe that this will completely compensate for the difference. Annotating the more frequent concepts of set b took longer on average than annotating the infrequent concepts of set a, with the Sapphire annotation system. As we have confirmed that there is no significant increase in disagreement when annotating up to five concepts simultaneously, multiple-concepts annotation appears to be the preferable method to maximise efficiency. The average annotation time per concept and image among all participants in multiple-concepts mode was 0.76 seconds. This is almost twice as fast as the 1.47 seconds that users needed on average to annotate in single-concept mode.

5.3 Summary

After conducting two user experiments to study manual annotation of visual documents, we are able to draw some valuable conclusions from the obtained data.

In the first experiment, we could confirm that using the keyboard allows faster annotation compared to using the mouse. After our second experiment, we cannot confirm any general dependency of inter-rater agreement on types of semantic concepts, such as vehicles versus settings and sites. However, we believe that individual concept specifications need to be
revisited because they seem to imply much ambiguity. Ideally, this is done with consideration of the annotators’ cultural and educational backgrounds. In both experiments, the concept Vegetation could be identified as a problematic concept based on low inter-rater agreement that caused poor classification performance. The same could be observed in the TRECVID annotation effort for the concepts Police/Security and Corporate Leader. Concepts such as these are particularly problematic because they can only be inferred by interpreting the visual content using specific knowledge or experience. This means that judgements on such concepts will most likely differ significantly between individuals, as has been shown in our experiments. Results of the second experiment also suggested that the concept Studio is difficult to annotate, while this was not the case in the TRECVID annotation. A possible explanation for this is the fact that users in the TRECVID 2005 annotation effort can be regarded as experts. These annotators handle television news footage frequently and have a clear idea of what a Studio shot looks like. In the second experiment that we conducted at RMIT University, users were not experienced in this context and perceived it to be very difficult to identify Studio shots. Given our observations of the TRECVID experiment, we believe that assessing the classification performance in combination with the annotation time yields a good indicator for concept difficulty.

It is difficult to interpret the results of both experiments to determine whether annotating in multiple-concepts mode is faster than annotating in single-concept mode. Our experiment at RMIT University suggests that annotating with multiple concepts is faster, while the evaluation of the TRECVID 2005 annotation showed an opposite trend. However, the results of our experiment at RMIT University are more reliable due to a setup that cancels out unwanted effects. To draw a conclusion, we must also consider the influence of the different annotation systems that were used. The Sapphire system displays one image per page; overhead is incurred due to navigating and page loading each time a new image is displayed. It is therefore not surprising that annotating in multiple-concepts mode is faster on this system than annotating in single-concept mode. The IBM system displays up to 200 images per page; each page is reloaded when a user switches the concept. This normally happens quickly because the images do not change, and are reloaded from the browser cache. Rendering the page, however, still requires extra time. The results collected with the IBM system are influenced by many factors and we observe that annotating with many concepts may be slower than annotating with one concept, but not necessarily so. As we could not confirm that annotating with five concepts simultaneously leads to lower agreement between raters, we conclude that annotating in multiple-concepts mode is the preferable strategy as long
as the number of concepts does not exceed the capacity of the annotators. The results of our experiment with the Sapphire annotation system suggest that five concepts can still be handled well by an average user. However, further experiments are necessary to establish a threshold for the maximum recommended number of concepts. In the experiment conducted at RMIT University, we find evidence that annotating frequent concepts requires more time than annotating infrequent concepts. This is not surprising as annotating concepts that occur more frequently requires more user interaction. The EVA system may have compensated for this effect by providing bulk annotation functionality.

For large collaborative annotation tasks, a server-based setup is beneficial because it allows centralised storage, and analysis of the annotation data during and immediately after the annotation. This allows for identification and possible correction of negative trends while the annotation is still in progress. It is preferable to display multiple images per page, and to allow customisation of parameters, such as the number of images per page and the image size. Our experiments showed that users make good use of such features, and that there is little influence of the number of displayed images on annotation time. For video annotation in particular, displaying multiple thumbnails per page yields the advantage of providing some temporal context if the key frames are shown in temporal order.
Chapter 6

Content-based Video Search

“All truths are easy to understand once they’re discovered; the point is to discover them.”

– Galileo

Effective video search requires the comprehensive analysis of the video content to build some form of index. Several fundamental processing steps, such as those discussed in the previous chapters, must be carried out before a truly multimodal index can be generated. Mapping low-level audiovisual features of video shots to high-level semantic concepts, enables concept-based search to retrieve sections of video that match a given information need. However, semantic concept terms that originate from automatic concept detection may not be sufficient to provide effective search. This is because automatically detecting semantic concepts is a very difficult task, and reliable detection can currently only be achieved for a very limited concept vocabulary. As a result, the spoken text from the video sound track is still an important source of semantic information [Hauptmann and Christel, 2004] for content-based video search.

A common approach to speech-based video search is to apply speech recognition techniques on the audio stream to convert any spoken information into text. This text can then be aligned to the segmented video, and used to build a text-based index for video search. A major advantage of this approach is that text-based search is a well understood and scalable technique. We can thus employ existing text search engines that use an inverted index for efficient querying and retrieval. A disadvantage of this method is that the visual content of the footage is not directly reflected in the index. Anything that is not referred to in the spoken text cannot be retrieved. Moreover, anything that is referred to but not visually present may lead to false positive detections. We discussed query expansion as an option
to address some of these problems in Section 2.4.2, but current research suggests that only integrating information from all modalities into a search approach can be successful.

In this chapter, we explore several aspects of video search using both speech text and semantic concept terms. In Section 6.1, we discuss different query expansion techniques for video search using only speech text. We demonstrate the topic dependency of each individual method, and propose a fusion approach to improve the retrieval effectiveness for a wider range of query topics. In Section 6.2, we present a fusion approach to combine semantic concepts with a speech-based inverted index for efficient video search. Instead of applying term expansion to the query, we expand the semantic concept terms in the index in this approach. We evaluate both methods using the TRECVID search test sets of previous years, and conclude the chapter with a discussion of our findings in Section 6.3.

6.1 Using Spoken Text for Video Search\(^1\)

As discussed in Section 2.4.1, speech-based video search is a popular approach for granular video retrieval at the shot or story level. Using time-aligned speech transcripts from Automatic Speech Recognition (ASR) or from Closed Captions (CC), footage can be represented as a collection of short text documents, and be searched with standard text-search engines. However, compared to text-document retrieval, video search with spoken text is more challenging because automatic speech recognition systems have a considerable word error rate [Hauptmann, 2005]. Another problem is that the spoken text does not necessarily reflect the visual content well, and we cannot retrieve visually present objects that the spoken text does not refer to. In addition, the appearance of events or objects of interest might not be exactly synchronised with their appearance in the spoken text. While this temporal misalignment can be addressed during the time-alignment step of the speech transcripts to the video shot documents [Nock et al., 2003b], it is more difficult to alleviate the mismatch in semantics between the spoken and the visual tracks. As a result, speech-based video retrieval tends to perform well in answering specific queries about named people, sites, or events, but usually fails at generic queries involving unnamed people, objects, settings, or events.

\(^1\)This work was conducted while the author was a visiting researcher at the IBM T. J. Watson Research Center in Hawthorne, NY, USA.
6.1. USING SPOKEN TEXT FOR VIDEO SEARCH

Figure 6.1: The IBM speech-based video retrieval system. This is built using the IBM Unstructured Information Management Architecture (UIMA) that includes several components developed by IBM Research for advanced text analytics.

6.1.1 Query Refinement

We discussed query refinement to address some of the problems in speech-based video retrieval in Section 2.4.2. Query refinement aims at improving the search results by automatically reformulating the original query text. The effects of this process are often that additional matches may be found — improving recall — by inserting additional query terms, or that the search results may be re-ranked which potentially improves precision. There are several different approaches to query reformulation, each with either a greater tendency to enhance recall or to enhance precision. The results usually depend on the query topics and on the collection. We are thus unlikely to find a single approach that improves both recall and precision for a broad range of query topics. To address this problem and improve overall effectiveness, we explore the effects of different query refinement techniques, and propose a combined approach.

6.1.2 Effectiveness of Query Refinement Techniques

We study the effects of three query refinement techniques when applied to speech-based video retrieval. These methods are explained in detail in Section 2.4.2 and are as follows:
Rocchio-based query refinement: Rocchio query refinement [Rocchio, 1976] is a pseudo-relevance feedback method in which $k$ representative terms for query expansion are selected from the top $N$ documents ranked highest by the original query.

Lexical affinity-based query refinement: While this approach is also based on pseudo-relevance feedback, it employs lexical affinities [Carmel et al., 2002] to select terms for query expansion. These are pairs of terms that frequently co-occur within close proximity of each other, but are not necessarily lexically related.

Semantic annotation-based refinement: In this method, the entire corpus is annotated and indexed with semantic categories. This annotation is also applied to the query text. The annotated query is passed to the search engine to be run against the annotated index.

We evaluate these three techniques with the speech-based video search system that is part of the IBM video retrieval system [Amir et al., 2004]. We consider only the speech-based part in this work to study the effects of query refinement in isolation. Our goal is to improve the existing search approach that did not use any query refinement [Amir et al., 2004]. We develop a new speech-based video search system using the IBM Unstructured Information Management Architecture (UIMA)$^2$ [Ferruci and Lally, 2004] and the JuruXML semantic search engine [Mass et al., 2002]. The JuruXML search engine is included in the UIMA Software Development Kit.$^3$ In addition, we use several UIMA components developed by IBM Research for advanced text analytics. These include the RESPORATOR (RESPOnse geneR-ATOR) system [Prager et al., 2000] and the PIQUANT question answering system [Carroll et al., 2004]. RESPORATOR is used extensively by the PIQUANT question answering system [Carroll et al., 2004] such that each query is analysed by PIQUANT and annotated with one or more semantic categories to disambiguate the query. As shown in Figure 6.1, the IBM system can perform the shot segmentation of the videos but we do not use this feature as part of our experiments. Instead, we use the common shot segmentation that was provided by NIST to keep our results comparable with others that are reported at TRECVID [Kraaij et al., 2004; 2006; Over et al., 2005; Smeaton et al., 2003].

We study the performance of the three query refinement approaches using different query topics. This is illustrated in Figure 6.2 using four sample topics taken from the

---

$^2$http://www.research.ibm.com/uima
$^3$http://www.alphaworks.ibm.com/tech/uima
6.1. USING SPOKEN TEXT FOR VIDEO SEARCH

Figure 6.2: Query-dependency of query expansion methods, illustrated on four example query topics from the TRECVID 2005 test collection. Performance is normalised so that the best-performing approach for the given topic has 100% score.

TRECVID 2005 search test set. This test set is described in detail below. The four queries represent different query types that frequently occur in the TRECVID search test corpora. As can be seen from this figure, the four topic types exhibit different performance patterns when using different query refinement techniques. In fact, for many topics, the best strategy is to not perform any query refinement. Such topics include named person queries, or difficult queries for which the pseudo-relevance assumption for the top documents does not hold.

We therefore propose a fusion approach to improve robustness and to combine the strengths of the individual approaches. In particular, we consider a global parameter-free fusion approach that does not require query-specific training. We use score averaging to combine the shot ranking scores as determined by the four individual search techniques. That is, we combine the three query refinement approaches with the query baseline that does not use any refinement.
### Table 6.1: The 24 queries that are specified in the TRECVID 2005 search test set, along with the query classes that we have manually assigned, based on the semantics of the query text.

<table>
<thead>
<tr>
<th>ID</th>
<th>Topic (raw query text)</th>
<th>Query class</th>
</tr>
</thead>
<tbody>
<tr>
<td>149</td>
<td>Find shots of Condoleezza Rice</td>
<td>Person-X</td>
</tr>
<tr>
<td>150</td>
<td>Find shots of Iyad Allawi, the former prime minister of Iraq</td>
<td>Person-X</td>
</tr>
<tr>
<td>151</td>
<td>Find shots of Omar Karami, the former prime minister of Lebanon</td>
<td>Person-X</td>
</tr>
<tr>
<td>152</td>
<td>Find shots of Hu Jintao, president of the People’s Republic of China</td>
<td>Person-X</td>
</tr>
<tr>
<td>153</td>
<td>Find shots of Tony Blair</td>
<td>Person-X</td>
</tr>
<tr>
<td>154</td>
<td>Find shots of Mahmoud Abbas, also known as Abu Mazen</td>
<td>Person-X</td>
</tr>
<tr>
<td>155</td>
<td>Find shots of a graphic map of Iraq, location of Baghdad marked</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>156</td>
<td>Find shots of tennis players on the court</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>157</td>
<td>Find shots of people shaking hands</td>
<td>Event/Activity</td>
</tr>
<tr>
<td>158</td>
<td>Find shots of a helicopter in flight</td>
<td>Object</td>
</tr>
<tr>
<td>159</td>
<td>Find shots of George W. Bush entering or leaving a vehicle</td>
<td>Person-X</td>
</tr>
<tr>
<td>160</td>
<td>Find shots of something on fire with flames and smoke visible</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>161</td>
<td>Find shots of people with banners or signs</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>162</td>
<td>Find shots of one or more people entering or leaving a building</td>
<td>Event/Activity</td>
</tr>
<tr>
<td>163</td>
<td>Find shots of a meeting with a large table and more than two people</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>164</td>
<td>Find shots of a ship or boat</td>
<td>Object</td>
</tr>
<tr>
<td>165</td>
<td>Find shots of basketball players on the court</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>166</td>
<td>Find shots of one or more palm trees</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>167</td>
<td>Find shots of an airplane taking off</td>
<td>Object</td>
</tr>
<tr>
<td>168</td>
<td>Find shots of a road with one or more cars</td>
<td>Setting/Scene</td>
</tr>
<tr>
<td>169</td>
<td>Find shots of one or more tanks or other military vehicles</td>
<td>Object</td>
</tr>
<tr>
<td>170</td>
<td>Find shots of a tall building</td>
<td>Object</td>
</tr>
<tr>
<td>171</td>
<td>Find shots of a goal being made in a soccer match</td>
<td>Event/Activity</td>
</tr>
<tr>
<td>172</td>
<td>Find shots of an office setting</td>
<td>Setting/Scene</td>
</tr>
</tbody>
</table>

6.1.3 Experiments and Evaluation

We conducted experiments using the TRECVID 2005 search test corpus, as described in Appendix C, and the query topics that were specified by NIST. These 24 text queries are shown in Table 6.1. The 2005 search test collection contains 140 broadcast news video clips in one of the three languages U.S. English, Arabic, or Chinese, pre-segmented into 45,765 shots by the Heinrich-Hertz-Institute [Petersohn, 2004]. The spoken information from the videos was automatically transcribed using state-of-the-art automatic speech recognition software and the Arabic and Chinese sources were machine translated into English [Over et al., 2005]. The resulting text transcripts, both native language and translated, were provided by NIST as part of the test collection.

For our evaluation, we manually define four query classes to group the 24 test set queries. The classes are chosen on the basis of preliminary experiments and our individual interpreta-
tion with the aim that queries within each group share one type of information need. These classes are as follows:

**Person-X:** Queries that request shots showing a specific named person, for example “Con- doleezza Rice”.

**Object:** Queries requesting shots that depict generic objects being visually present.

**Setting/Scene:** This type of query request shots of a specific setting. This may involve people or activities, but the focus is on the presence of a visually distinctive environment.

**Event/Activity:** Queries that request an event or an activity that usually involves people. However, no specific people are required and the focus is on the event or activity.

However, a manual query class definition of this form is likely to be sub-optimal because it is biased by our interpretation of the semantics in the queries. We believe that further investigation of mining query classes on a larger scale is necessary to develop a more generalised classification scheme. In our query classification, we further require that each query can only belong to one query class so that we can provide an unambiguous mapping of queries to query classes. Each query was then manually assigned to one of our four query classes.

Each video in the test collection has a corresponding speech transcript obtained through automatic speech recognition, as well as machine translation into English for the Arabic and Chinese sources. This data is provided by NIST to ensure a common test-bed among all TRECVID participants. In all our search methods, we pre-process queries to perform part-of-speech tagging, and retain only nouns using the components of the UIMA framework that we have described previously. Our preliminary experiments have shown that this is the preferable approach as the nouns best convey the information need for visual information retrieval purposes. Our query processing interface also automatically strips off the request phrase “Find shots of”. When indexing the text-document corpus, we perform stemming using the well-known Porter [1980] algorithm. The JuruXML search engine also natively identifies phrases in the form of lexical affinities, and uses them to resolve ambiguities and to obtain term occurrence statistics for the collection.

---

4Results presented here may differ from previously published results [Volkmer and Natsev, 2006] because we did not enforce mutual exclusiveness in our prior work.
As discussed in Section 2.4.2, the Rocchio- and the LA-based query refinement approaches that use pseudo-relevance feedback require manual tuning of three parameters. These parameters are as follows:

**Number of pseudo-relevant documents:** The number $N$ of top-ranked documents returned by the original query that are to be considered relevant.

**Maximum number of terms to add:** The number $k$ of terms that the system is allowed to add when expanding the original query.

**Weight of additional terms:** The weight $\beta$ to be assigned to the expanded terms as opposed to the original terms. The parameter $\beta$ is specified such that $0 \leq \beta \leq 1$, where 1 that the added terms are assigned the same weight as the original query terms.

To determine optimum settings for these parameters, we conducted training runs on the TRECVID 2003 search test set and queries, described in Appendix C. However, we do not elaborate on these training experiments here, as we are not focusing on optimising the performance of the individual query expansion approaches. Interestingly, we identified the optimal number of pseudo-relevant documents to be $N = 30$ for lexical-affinity based expansion, and to be $N = 12$ for the Rocchio method in our experiments. This difference may be explained by the LA-based expansion method being less prone to causing topic drift [Carmel et al., 2002] than the Rocchio method. Due to the different term selection technique in LA-based query expansion, this method was more effective with larger values for $N$, while the Rocchio method already caused considerable topic drift for $N > 12$ in our preliminary experiments. Conversely, the optimal settings for both the maximum number of terms to add $k$ and the weight for added terms $\beta$ was consistent between both approaches. We determined $k = 30$ and $\beta = 1.0$ as the optimal values for both methods.

These parameters are likely to be sub-optimal for the 2005 corpus. While the 2003 and the 2005 TRECVID collections are both based on broadcast news, they differ in other aspects, such as including different channels with different production rules, and different quality of the speech transcripts. For most of the clips of the 2003 collection, NIST provided text transcripts from closed captions in addition to the ASR sources. The CC transcripts usually yield better search performance as their word error rate is much below that of the ASR transcripts. When available, we used the CC transcripts instead of the ASR transcripts with the 2003 corpus [Smeaton et al., 2003]. Moreover, the 2005 test set contains 43.2 hours
6.1. USING SPOKEN TEXT FOR VIDEO SEARCH

<table>
<thead>
<tr>
<th>Query Refinement/Expansion Method</th>
<th>Training Set TRECVID 2003</th>
<th>Test Set TRECVID 2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>No refinement</td>
<td>0.0831</td>
<td>0.0558</td>
</tr>
<tr>
<td>Semantic refinement</td>
<td>0.1237</td>
<td>0.0546</td>
</tr>
<tr>
<td>LA-based expansion</td>
<td>0.1275</td>
<td>0.0578</td>
</tr>
<tr>
<td>Rocchio expansion</td>
<td>0.1291</td>
<td>0.0413</td>
</tr>
</tbody>
</table>

Table 6.2: Mean average precision scores of the text search baseline (no refinement) and three query expansion approaches, evaluated on two different corpora and two sets of search topics. Parameters were tuned to optimise TRECVID 2003 performance and were applied blindly on TRECVID 2005 data and topics.

of Arabic and 52.3 hours of Chinese sources, as well as 74.2 hours of U.S. English sources. As discussed in Section 2.4.1, machine translation increases the word error rate, further reducing the quality of the speech transcripts. As a result, search results using the machine translated speech sources are usually poor compared to results using native English sources. Due to the lack of other sources, we used the machine translated ASR transcripts provided with the 2005 test set. To evaluate performance, we conducted blind runs on the TRECVID 2005 test set using the 24 search topics as specified for the 2005 search task evaluation. As is common for search evaluations such as TREC and TRECVID, we use precision and recall to measure the effectiveness of our approaches, as defined in Section 2.4.4. In particular, we use Average Precision (AP) for ranked results in response to a single query and Mean Average Precision (MAP) for the results of multiple queries. Table 6.2 shows a comparison of the MAP scores for the three query refinement approaches and the baseline (no query refinement), as evaluated in training runs on the 2003 collection and in blind runs on the 2005 collection.

Although the two sets of results are based on different sets of query topics, and are therefore not directly comparable, we still observe a significant performance loss on the 2005 corpus. This is most likely due to the poor quality of the machine translated non-English sources, and to sub-optimal parameters for the 2005 data. More interesting is the discrepancy in relative performance of different approaches on the two corpora. Our results on the 2003 collection show that query expansion can yield significant (50%) improvements when properly tuned. The opposite is true on the noisier 2005 data — only one of the query expansion approaches outperforms the no-expansion baseline.

138
Figure 6.3: Query-class specific performance evaluation, based on our manually defined query classes, with the TRECVID 2005 test set. Query expansion is highly topic-dependent and no single method is likely to outperform the others on every topic. Person-X queries often work better without any query refinement.

Next, we analyse the performance when grouping the query topics into the previously defined query classes. Figure 6.3 shows the query-class specific performance of the three query expansion approaches and the baseline on the TRECVID 2005 test set. This figure confirms that query expansion is highly topic-dependent, and that no single method is likely to outperform the others on every topic. It also yields a possible explanation why query expansion hurts overall performance for two of the methods. Since the overall MAP score is influenced mostly by top-performing queries, the Person-X query class dominates all other query classes due to the much higher scores it generates. Any approach that does not perform well on Person-X topics is therefore likely to have poor overall performance. Incidentally, Person-X queries are often processed with best average precision when no query refinement is applied.

As no single approach works best for all topics, we aimed at minimising the query-dependency by combining our query expansion techniques with the baseline approach. We use a score-averaging fusion scheme to combine all four methods. Specifically, we apply global parameter-free score-averaging in which the final result score of an item is generated
6.1. USING SPOKEN TEXT FOR VIDEO SEARCH

<table>
<thead>
<tr>
<th>Query Class</th>
<th>No query Expansion</th>
<th>Query Expansion Fusion</th>
<th>Average (gain)</th>
<th>Oracle (gain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person-X (7)</td>
<td>0.1133</td>
<td>0.1238 (9%)</td>
<td>0.1348 (19%)</td>
<td></td>
</tr>
<tr>
<td>Object (5)</td>
<td>0.0367</td>
<td>0.0452 (23%)</td>
<td>0.0622 (69%)</td>
<td></td>
</tr>
<tr>
<td>Setting/Scene (9)</td>
<td>0.0171</td>
<td>0.0234 (37%)</td>
<td>0.0260 (52%)</td>
<td></td>
</tr>
<tr>
<td>Event/Activity (3)</td>
<td>0.0699</td>
<td>0.0594 (-15%)</td>
<td>0.0752 (8%)</td>
<td></td>
</tr>
<tr>
<td>All Topics (24)</td>
<td>0.0558</td>
<td>0.0617 (11%)</td>
<td>0.0745 (34%)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: Query-class specific mean average precision scores for no-expansion baseline, average fusion-based query expansion, and an Oracle method with test set-optimised fusion parameters.

by computing the unweighted average of all result scores from the individual approaches for that item. If an item does not exist in the result list of an approach, we assume the result score to be 0 (zero) for this approach.

To identify the potential gains when applying a query-dependent fusion scheme, we define an Oracle fusion. For the Oracle evaluation, we consider five different score normalisation methods, along with three non-weighted fusion methods (average, maximum, and product), and then choose the optimal combination for each individual query as observed on the 2005 test set. Naturally, such an approach cannot be implemented without knowing the result in advance, but this test serves as a measure of the maximum potential performance gains with optimally tuned query-specific fusion parameters.

The results of the combination hypothesis approach are listed in Table 6.3. The simple average fusion approach does improve robustness, as it outperforms the baseline for three out of four query classes. It leads to 11% overall improvement and a mean average precision within 10% of the best performance (0.067) reported for text-based automatic search in TRECVID 2005 [Over et al., 2005]. Moreover, the Oracle method demonstrates significant potential gains for all query classes, ranging from 8% to 69%, with an overall improvement of 34% over all topics. This attests to the promise of this combination hypothesis approach for query expansion, and shows that further research to design query-specific approaches could significantly improve the results. Based on the Wilcoxon [1945] signed rank test, both the results for average fusion and Oracle fusion are statistically significant at the 5% level.

As our results show, the corpus-based query expansion techniques that we tested in our experiments can improve the effectiveness of speech-based video retrieval. We achieve
significant improvements through a fusion approach of several query expansion methods, but this technique is clearly limited as we consider only the speech modality of the video. Visually present objects, people, settings, or events that are not mentioned in the spoken information cannot be retrieved. The key to effective content-based video search is processing and incorporating information from multiple modalities, in particular the visual modality. In the following section, we discuss a fusion approach combining spoken and visual information to implement a multimodal search technique using a free-text query interface.

6.2 A Fusion Approach to Content-based Video Search

We aimed at maintaining a free-text query interface that does not require the user to provide visual query examples, but incorporates semantic information extracted from the visual domain in the search approach. In speech-based video retrieval, we observe that we can successfully retrieve relevant video segments as long as the object of interest is present in the spoken information. Our hypothesis is therefore that it should be possible to add terms describing the visual content to the speech transcripts, and so enable retrieval of objects that are visually present but not mentioned in the spoken text.

Our baseline search approach includes only spoken text; we generate a shot-aligned text document collection as previously described, and build an inverted index for this collection. We then add semantic concept terms to the shot-aligned speech transcripts that describe the visual content of the video shot. These terms are taken from a set of 101 automatically detected visual concepts [Snoek et al., 2006c] that were made available for the TRECVID test collections for 2005 and 2006. Besides adding the concept terms, we also experimented with expanding these terms based on lexical semantic referencing before adding them to the index. The 101 concept terms that were provided covered only a narrow field of topics and we wanted to test whether we can improve the overall retrieval performance by including lexically and semantically related terms in addition to the detected concepts. As a third extension to our baseline technique, we consider a query-dependent method that invokes a specialised retrieval method upon semantic analysis of the given query. These approaches are different from those of Haubold et al. [2006] that we explained in Section 2.4.2 as we do not use a visual retrieval step. Instead, our approaches use only one text-based search run.

We evaluate our techniques in training experiments on the TRECVID 2005 test set and in blind runs on the TRECVID 2006 test set. In the remainder of this chapter, we outline our system set-up, describe the experiments that we conducted and discuss outcomes in detail.
6.2. A FUSION APPROACH TO CONTENT-BASED VIDEO SEARCH

Figure 6.4: A screenshot of the Sapphire search engine web front-end. It allows efficient text-based video search. Results are visualised for each returned shot using one representative frame.

6.2.1 The Sapphire Search System

We developed the Sapphire search engine to provide fully automatic, text-based video search, while allowing to add semantic key words and perform automatic query analysis. Figure 6.4 shows a screenshot of the Sapphire search engine web interface. Query results are visualised by showing one representative frame for the video segments that are considered relevant. In Sapphire, we use the Indri text search engine and text indexer [Abdul-Jaleel et al., 2004; Metzler and Croft, 2004] that are part of the Lemur Toolkit\(^5\) developed at Carnegie Mellon University and the University of Massachusetts, Amherst. In all our experiments, we apply the Indri built-in Porter [1980] stemming algorithm at indexing time. We also use the OpenNLP\(^6\) natural language processing package for text analytics and WordNet\(^7\) for lexi-

\(^5\)http://www.lemurproject.org
\(^6\)http://opennlp.sourceforge.net
\(^7\)http://wordnet.princeton.edu
Figure 6.5: The Sapphire video retrieval system without the result visualisation engine. This uses the Lemur Toolkit for text indexing and search, and the OpenNLP package for text analytics.

cal semantic referencing, as described below. Figure 6.5 shows an overview of the Sapphire search system, including those parts of the system that we use to add the semantic concept terms, and to process the incoming queries.

6.2.2 Adding Semantic Concept Terms

The MediaMill team at the University of Amsterdam,\(^8\) a regular TRECVID participant, provided semantic concept annotations for the TRECVID 2005 and TRECVID 2006 test corpora based on automatic detection of 101 concepts [Snoek et al., 2006c]. In particular, we leveraged the output of their Experiment 1 that was designed to classify shots based on only visual low-level features. The annotation is provided as lists of shot references, one for each concept, in which all shots are assigned a score from the Support Vector Machine (SVM) automatic concept detector that represents the estimated probability that this shot depicts the given concept. These scores are not normalised and are reported as values between 0.0 and 1.0. This means that the highest ranked shot may have a score lower than 1.0 and the lowest ranked shot may have a score higher than 0.0. Moreover, the highest and lowest scores differ substantially across all concepts due to different classification performances of

\(^8\)http://www.science.uva.nl/research/mediamill

143
the detectors between concepts. For some concepts, the concept detector scores are generally very low which is an indicator that the classification performance for these concepts may not be very reliable.

To be able to sensibly decide when to add a semantic concept term to a shot, we must make an assumption of the prevalence of the concept throughout the collection. We estimate the prevalence of each concept in the test collection based on the concept prevalence in the development collection, the collection that has been used to train the semantic concept detectors of the MediaMill team. This is possible because the MediaMill team also provided the manual annotation data on the common TRECVID 2005 development corpus that they used as the ground-truth during training. This ground-truth includes the annotations from the TRECVID 2005 annotation forum for 39 semantic concepts, with added annotations for 62 further concepts by the MediaMill team.

First, we assume approximately equal distributions of all concepts between the development and the test collection. Given that the development and test collections are highly correlated in terms of their semantics, we believe that this is a reasonable assumption. Second, as the collection sizes differ between development and test collections, we extrapolate the known prevalence of each concept in the development collection to the test collection. This results in an expected number of relevant shots $n$ in the test collection for each concept. We can then define a threshold for each concept that we apply to the detector probability scores provided with the automatic annotations. For this, we rank all shots such that the shot with the highest score is ranked first. To add semantic concept terms to the shot-aligned text documents in our collection, we add the concept term to the spoken text transcript of the $n$ top-ranked shots from the automatically detected concepts, where $n$ is the number of expected relevant shots for the given concept.

However, as previously described, the probability scores may be very low for some concepts. If we added the expected number of $n$ shots in such cases, we would likely add many false positive detections. To avoid adding too many false positive detections, we specify a minimum probability score that must be satisfied globally. As we discuss below, we conducted training experiments using the TRECVID 2005 search test set to determine the optimal threshold for this minimum score.

Adding semantic concept terms, however, will only allow effective retrieval of visual content if the query contains a specific term exactly as it was defined in the vocabulary that

---

was used during automatic concept detection. To enable successful retrieval using semantically related terms, we experimented with expanding concept terms through lexical semantic referencing.

6.2.3 Semantic Concept Term Expansion

We leverage the WordNet [Miller, 1995] lexical semantic referencing system to perform this term expansion. For each semantic concept term that is to be added to the index, we query WordNet and select additional terms from those that are returned. We explored the following strategies for this term expansion technique:

**Hyponym expansion:** A hyponym is a word or a term that denotes a subordinate of another given word. For example, “Dog” is a hyponym of “Animal”. For hyponym expansion, we retrieve the corresponding hyponyms from the WordNet database for each concept term associated with the shot, and then add the original concept term and its hyponyms to the speech transcript of the shot.

**Hypernym expansion:** A hypernym is the opposite of a hyponym and a term that denotes a superordinate, or a superclass, of another word. For example, “vehicle” is a hypernym of “car”. Accordingly, for hypernym expansion, we retrieve WordNet hypernyms associated with each concept term and add the original concept term and its hypernyms to the speech transcript of the shot.

**Synonym expansion:** Synonyms are two or more terms that can interchangeably be used to describe the same concept or object. A synonym is an alternative term, for example, “lawyer” and “attorney” are synonyms. To perform synonym expansion, we add the original concept term and its synonyms to the transcript of the shot.

**Meronym expansion:** A meronym is a term that describes a concept that is part of another concept. For example, “knob” is a meronym of “door”. We perform meronym expansion by adding the original concept term and its meronyms to the transcript of the shot.

**Hyponym and Hypernym expansion:** In this approach, we combine the first two strategies and retrieve both hyponyms and hypernyms from WordNet for each concept term, and add these to the speech transcript alongside the concept terms.

With all term expansion strategies, we observed significant improvements over using only semantic concept terms during training. Moreover, the different strategies exhibit different
tendencies to topic drift, and are therefore not equally effective. We also noticed that, as with query refinement and expansion techniques, the results are query-dependent. In particular, we observed the best performance for queries concerning named persons, locations, or named organisations when using only the speech transcripts without semantic concept terms in any form. To cater for this, we apply a query-dependent approach that includes analysing the query text with a named-entity detector. This allows us to identify query terms that indicate whether a query concerns a person, a location, or a known organisation. Based on this named-entity detection, our system can invoke a search using a specialised inverted index.

6.2.4 Query Preparation and Analysis

The query processing interface uses the named-entity detector and the part-of-speech tagger of the OpenNLP Tools.\textsuperscript{10} These are model-based detectors that use a maximum-entropy machine-learning system.\textsuperscript{11}

Our system processes all queries for part-of-speech tagging after removing the request phrase, “Find shots of...”, that is common for queries specified by NIST for the TRECVID evaluation. Part-of-speech tagging allows us to retain only the nouns of the query text. In preliminary experiments, this technique has shown to yield generally better results than using the full query text, due to the TRECVID queries and evaluation targeting visual information. Nouns tend to convey such visual information needs best. Our preliminary experiments suggested that results can be improved in some cases when retaining nouns and verbs, however, overall results were best when we used only nouns.

As our training runs suggested that best retrieval performance can be achieved with a query-dependent approach, we classify queries based on automatic named-entity detection. If it can be determined that the query contains a specific named person, location, or organisation, the system invokes a search based on the speech transcripts only. For all other queries, it uses an index built from speech transcripts, with semantic concept terms and relevant hyponyms added. We chose this type of index because, during training, we observed best results for non-named entity related queries with the hyponym expanded index. Due to time constraints and a lack of adequate training corpora, we did not train models specifically for our purposes, but instead rely on the default models included in the OpenNLP package.

\textsuperscript{10}http://opennlp.sourceforge.net
\textsuperscript{11}http://maxent.sourceforge.net
While the results are generally very good, we observe occasional detection errors that might be avoided with more suitable models.

### 6.2.5 Training Experiments with the TRECVID 2005 Test Set

We conducted training experiments with the TRECVID 2005 search test set and queries. First, we determine optimal parameters for the Indri built-in pseudo-relevance feedback feature [Lavrenko and Croft, 2001] that we enabled in all our runs. These parameters were the number of top-ranked documents to consider \( N = 10 \), the maximum number of terms to add \( k = 20 \), and the weight of the added terms \( \beta = 1.0 \). However, the focus of our training experiments is selecting an optimal confidence threshold \( t \) when adding semantic concept terms and exploring the effects of different semantic term expansion techniques in our indexing approach. WordNet usually returns a hierarchy of related terms; this hierarchy can be very deep, depending on how many related terms are found in the database for a specific word. Each level of this hierarchy can contain one or more semantically related terms. The deeper terms are positioned in this hierarchy, the more distant the relationship to the original. This means that the risk of topic drift increases when adding terms from the deeper levels of the hierarchy. In our experiments, we do not explicitly limit the number of terms to be added, but we limit the maximum depth in the hierarchy that our algorithm descends to when adding terms.

Figure 6.6 shows the mean average precision results of the different term expansion strategies when varying the expansion depth \( D \). These results were obtained with a fixed confidence threshold of \( t = 0.15 \); the baseline performance of 8.47\% mean average precision is observed for \( D = 0 \), that is when not adding any terms besides the original semantic concept terms. Meronym expansion yields the smallest overall improvement over this baseline, with a maximum of 9.49\% mean average precision at \( D = 8 \). Synonym expansion performance peaks at \( D = 2 \) with a mean average precision of 10.78\% and remains constant thereafter. This is because the maximum level found for all our terms when using synonym expansion is two, and so increasing \( D \) beyond two has no effect. Hypernym expansion performs slightly better, with a best mean average precision of 10.83\% at \( D = 5 \). The best term expansion strategy in these experiments is hyponym expansion with the maximum expansion level set to \( D = 5 \): we observe a mean average precision of 11.03\%, a 30\% improvement over the baseline. In addition, we experimented with a combined approach of the two best-performing methods, using both hypernyms and hyponyms to expand the semantic concept terms. As can be
6.2. A FUSION APPROACH TO CONTENT-BASED VIDEO SEARCH

Figure 6.6: Mean average precision in training runs on the TRECVID 2005 test set when varying the maximum expansion depth when expanding semantic concept terms using WordNet.

deduced from the graphs shown in Figure 6.6, this combination yields its best mean average precision of 10.91% at $D = 2$.

Figure 6.7 illustrates the effect of varying the minimum confidence score $t$ for our best-performing retrieval strategies on the TRECVID 2005 test set. The results shown in this graph are obtained using a maximum expansion depth of $D = 5$ for all strategies that use WordNet term expansion. For comparison, we include results for the baseline approach that uses no semantic concept terms, and relies only on spoken text. The graph in Figure 6.7 also includes the results for our query-dependent technique, where we use a named-entity finder to flag queries that concern a specific person, a specific location, or a named organisation. For such queries, the system invokes a search that uses only the spoken text. For all other queries, it uses the search approach that includes the semantic concept terms expanded with hyponyms. As can be seen from Figure 6.7, all approaches that use term expansion achieve the best performance for $t = 0.15$. Our best-performing training run uses the query-dependent search, achieving a mean average precision of 11.67%; this represents an improvement of over 60% over the speech-only baseline. We confirmed statistical signifi-
Figure 6.7: Performance of our system in training runs on the TRECVID 2005 test set when varying the minimum confidence threshold \( t \) for concept selection. Nearly all approaches yield the best performance for \( t = 0.15 \). The query-dependent approach achieved the best overall mean average precision.

cance of the improvements at the 5%-level for all the methods that we tested in our training experiments using the Wilcoxon [1945] signed rank test.

6.2.6 TRECVID 2006 Experiments

We participated in the TRECVID 2006 search task in the category of fully automatic search systems. Each group was allowed to submit up to six search runs in this category; all queries were required to be processed automatically by the search system without any user interaction. Similar to the TRECVID 2005 test corpus, the TRECVID 2006 test corpus consists of television news footage recorded from recent broadcasts in U.S. English, Chinese, or Arabic. The University of Pennsylvania Linguistic Data Consortium (LDC)\(^{12}\) provided speech transcripts for all videos of this collection based on closed captions and automatic speech recognition. Closed captions and ASR transcripts are available for most of the U.S. English

\(^{12}\text{http://www.ldc.upenn.edu}\)
6.2. A FUSION APPROACH TO CONTENT-BASED VIDEO SEARCH

sources, and machine translated ASR transcripts are available for the Chinese and Arabic sources.

The search test set in TRECVID 2006 is substantially larger than the collections of previous years [Kraaij et al., 2004; 2006; Over et al., 2005] and consists of 259 video clips with a total duration of nearly 160 hours. This includes 82.6 hours of Arabic, 29.7 hours of Chinese, and 46.3 hours of U.S. American footage. As with the collections of previous years, the news sections are interspersed with sections of advertisement and entertainment programming. However, the 2006 corpus contains recordings from a greater variety of television channels, some of which had not appeared in previous test corpora [Kraaij et al., 2006]. The test set contains a common shot boundary reference, provided by the Fraunhofer Institute [Petersohn, 2004], that we use to segment the speech transcripts based on the given timing, and align them with the video shots. We have previously observed generally better retrieval performance for text-based search when using closed caption text rather than ASR transcripts, as the former has significantly fewer errors. For this reason, we use closed captions where available, and fall back to using ASR transcripts for clips that have no provided closed captions.

Table 6.4 shows the 24 queries that NIST defined for the 2006 search evaluation along with the automatic classifications by our query processing stage. Query 179 (“Saddam Hussein”) is falsely detected as a location, and query 194 (“Condoleezza Rice”) is not detected as a named person by our query classifier. The queries were designed to put focus on visual content rather than specific topics for which speech-based search approaches tend to perform well [Kraaij et al., 2006]. In addition, many of the queries request visual concepts, such as a person or an object, in combination with an event or activity that extends over time. Most of the visual search approaches presented at TRECVID use one static key frame per shot [Kraaij and Over, 2006] and are less effective for detecting events. This also concerns the MediaMill system and the 101 automatically detected concepts that we utilise in our approach. We thus expected substantially poorer results on the TRECVID 2006 test set compared to the TRECVID 2005 test set.

We performed six blind runs on the TRECVID 2006 test set that resemble our five best training runs that we have described above and the baseline approach. Accordingly, the six test runs used hyponym expansion, hypernym expansion, and the combined approach that uses both for term expansion. We thus define the following six experimental runs:

**Run 1:** In this run, we select the concept terms to be added for each shot as previously described, and retrieve the corresponding hyponyms from WordNet for each concept.
### Table 6.4: The 24 queries that are specified for the TRECVID 2006 search test set, along with the query type that is automatically detected by our query processing stage, based on output of the OpenNLP named-entity detector (we do not include the common request phrase “Find shots of” in this table). As can be seen, queries 179 and 194 are not classified correctly.

<table>
<thead>
<tr>
<th>ID</th>
<th>Topic</th>
<th>Query type</th>
</tr>
</thead>
<tbody>
<tr>
<td>173</td>
<td>One or more emergency vehicles in motion</td>
<td>Other</td>
</tr>
<tr>
<td>174</td>
<td>A view of one or more tall buildings and the top story visible</td>
<td>Other</td>
</tr>
<tr>
<td>175</td>
<td>One or more people leaving or entering a vehicle</td>
<td>Other</td>
</tr>
<tr>
<td>176</td>
<td>One or more soldiers, police, or guards escorting a prisoner</td>
<td>Other</td>
</tr>
<tr>
<td>177</td>
<td>A daytime demonstration/protest with at least part of one building visible</td>
<td>Other</td>
</tr>
<tr>
<td>178</td>
<td>US Vice President Dick Cheney</td>
<td>Person</td>
</tr>
<tr>
<td>179</td>
<td>Saddam Hussein with at least one other person’s face</td>
<td>Location</td>
</tr>
<tr>
<td>180</td>
<td>Multiple people in uniform and in formation</td>
<td>Other</td>
</tr>
<tr>
<td>181</td>
<td>US President George W. Bush, Jr. walking</td>
<td>Person</td>
</tr>
<tr>
<td>182</td>
<td>One or more soldiers or police with one or more weapons and military vehicles</td>
<td>Other</td>
</tr>
<tr>
<td>183</td>
<td>Water with one or more boats or ships</td>
<td>Other</td>
</tr>
<tr>
<td>184</td>
<td>One or more people seated at a computer with display visible</td>
<td>Other</td>
</tr>
<tr>
<td>185</td>
<td>One or more people reading a newspaper</td>
<td>Other</td>
</tr>
<tr>
<td>186</td>
<td>A natural scene with, for example, fields, trees, sky, lake, mountain, etc.</td>
<td>Other</td>
</tr>
<tr>
<td>187</td>
<td>One or more helicopters in flight</td>
<td>Other</td>
</tr>
<tr>
<td>188</td>
<td>Something burning with flames visible</td>
<td>Other</td>
</tr>
<tr>
<td>189</td>
<td>A group of at least four people in suits, seated, with at least one flag</td>
<td>Other</td>
</tr>
<tr>
<td>190</td>
<td>At least one person and at least 10 books</td>
<td>Other</td>
</tr>
<tr>
<td>191</td>
<td>At least one adult person and at least one child</td>
<td>Other</td>
</tr>
<tr>
<td>192</td>
<td>A greeting by at least one kiss on the cheek</td>
<td>Other</td>
</tr>
<tr>
<td>193</td>
<td>Smokestacks, chimneys, or cooling towers with smoke/vapour coming out</td>
<td>Other</td>
</tr>
<tr>
<td>194</td>
<td>Condoleezza Rice</td>
<td>Other</td>
</tr>
<tr>
<td>195</td>
<td>One or more soccer goalposts</td>
<td>Other</td>
</tr>
<tr>
<td>196</td>
<td>Scenes with snow</td>
<td>Other</td>
</tr>
</tbody>
</table>

term associated with that shot. We then add the selected concept terms and their hypernyms to the speech transcript of the shot. The search operation uses an inverted index of the resulting text documents.

**Run 2:** This takes a query-dependent approach. Using the OpenNLP named-entity finder, our system analyses the query text and checks whether the query is about a specific person, a specific location, or an organisation. If it is, the search operation uses only the speech transcripts. If not, the search is performed with the same inverted index as Run 1.

**Run 3:** Here, we retrieve WordNet hypernyms associated with each concept term. We add the selected concept terms and their hypernyms to the speech transcript of the shot.
6.2. A FUSION APPROACH TO CONTENT-BASED VIDEO SEARCH

**Run 4:** In this run, we combine the strategies of Run 1 and Run 3. We retrieve both hyponyms and hypernyms from WordNet for each concept term, and add these to the speech transcript alongside the concept terms.

**Run 5:** In this run, we do not use lexical semantic referencing, and add only the selected concept terms to the speech transcript of the shot.

**Run 6:** This is our baseline run that uses only an inverted index based on the provided speech transcripts.

We set the minimum confidence threshold to $t = 0.15$ in those runs that use semantic concept terms. The maximum depth for concept term expansion with WordNet was set to $D = 5$ in all runs that use semantic referencing.

As described in Section 2.4.4, NIST distinguishes between the three system types A (using only NIST provided data), B (using NIST provided and additional data), and C (any system that is not A or B). Apart from our baseline run, which uses only the speech transcripts that are provided as part of the test collection, our runs are classified as type B because we use the semantic concept annotation that the MediaMill team has provided [Snoek et al., 2006c]. The results of our six submitted runs in terms of mean average precision are shown in Table 6.5.

In all runs, we have been able to achieve small improvements over the baseline. However, the results are generally substantially weaker than those that we obtained during training on the TRECVID 2005 set. In contrast to our observations during training, our best run in terms of mean average precision is Run 5 with a 12.8% improvement over the baseline. This run does not use term expansion and only adds semantic concept terms to the speech-based index.

We also evaluate the precision within the first 20 returned results (P@20) because the web interface of our search system shows 20 results per page. These results are shown in Table 6.6. Compared to the mean average precision results, we observe more substantial improvements in the precision at 20 returned results. For example, Run 1 shows a 17.5% improvement over the baseline. This shows that our approach causes a re-ranking of the results so that more relevant results appear within the top-ranked results.

However, according to the Wilcoxon [1945] signed rank test, the improvements that we observe for precision at 20 results and for mean average precision are not statistically significant. We are not able to replicate the improvements of our training experiments in the
### Table 6.5: Average precision for individual query for all our submitted runs, along with the Mean Average Precision (MAP) for each run. The best-performing approach is highlighted in bold for each query. We were not able to translate the good training results into significant improvements on the TRECVID 2006 test set. Mean average precision remains at a low level, with small improvements over the baseline.

TRECVID 2006 test runs. This may be largely due to substantial errors introduced by the automatic concept detection. A general observation during TRECVID 2006 was that automatic concept detection with visual features was less effective compared to TRECVID 2005, mostly because over 32% of the TRECVID 2006 collection is footage that is not represented in the development collection [Smeaton and Ianeva, 2006]. The proportion of non-English sources in the 2006 collection is substantially larger than in the 2005 collection, and so speech-based search techniques suffer more from the poorer quality of machine translated text sources. This has a direct impact on the search results, which are on average substantially lower than the results reported at TRECVID 2005 [Smeaton and Ianeva, 2006]. The proportion of rele-

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Run 6 (Baseline)</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>173</td>
<td>0.0026</td>
<td>0.0016</td>
<td>0.0016</td>
<td>0.0020</td>
<td>0.0029</td>
<td>0.0020</td>
</tr>
<tr>
<td>174</td>
<td><strong>0.0003</strong></td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td><strong>0.0003</strong></td>
</tr>
<tr>
<td>175</td>
<td>0.0005</td>
<td>0.0000</td>
<td>0.0000</td>
<td><strong>0.0010</strong></td>
<td>0.0011</td>
<td>0.0001</td>
</tr>
<tr>
<td>176</td>
<td>0.0007</td>
<td>0.0008</td>
<td>0.0008</td>
<td>0.0008</td>
<td><strong>0.0009</strong></td>
<td>0.0005</td>
</tr>
<tr>
<td>177</td>
<td>0.0396</td>
<td>0.0274</td>
<td>0.0274</td>
<td>0.0332</td>
<td>0.0323</td>
<td><strong>0.0402</strong></td>
</tr>
<tr>
<td>178</td>
<td><strong>0.2264</strong></td>
<td>0.2170</td>
<td><strong>0.2264</strong></td>
<td>0.2225</td>
<td>0.2077</td>
<td>0.2250</td>
</tr>
<tr>
<td>179</td>
<td><strong>0.1290</strong></td>
<td>0.0989</td>
<td><strong>0.1290</strong></td>
<td>0.1024</td>
<td>0.0920</td>
<td>0.1233</td>
</tr>
<tr>
<td>180</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0003</td>
<td><strong>0.0005</strong></td>
<td>0.0000</td>
</tr>
<tr>
<td>181</td>
<td>0.00008</td>
<td><strong>0.0131</strong></td>
<td>0.0008</td>
<td>0.0104</td>
<td>0.0073</td>
<td>0.0025</td>
</tr>
<tr>
<td>182</td>
<td><strong>0.0327</strong></td>
<td>0.0078</td>
<td>0.0078</td>
<td>0.0072</td>
<td>0.0039</td>
<td>0.0162</td>
</tr>
<tr>
<td>183</td>
<td><strong>0.0212</strong></td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0166</td>
<td>0.0102</td>
<td>0.0201</td>
</tr>
<tr>
<td>184</td>
<td>0.0078</td>
<td><strong>0.0092</strong></td>
<td><strong>0.0092</strong></td>
<td>0.0089</td>
<td>0.0111</td>
<td>0.0082</td>
</tr>
<tr>
<td>185</td>
<td>0.0087</td>
<td>0.0415</td>
<td>0.0415</td>
<td>0.0284</td>
<td><strong>0.0645</strong></td>
<td>0.0182</td>
</tr>
<tr>
<td>186</td>
<td>0.0017</td>
<td>0.0099</td>
<td>0.0099</td>
<td>0.0009</td>
<td><strong>0.0117</strong></td>
<td>0.0048</td>
</tr>
<tr>
<td>187</td>
<td>0.0351</td>
<td>0.0309</td>
<td>0.0309</td>
<td>0.0327</td>
<td>0.0320</td>
<td><strong>0.0355</strong></td>
</tr>
<tr>
<td>188</td>
<td>0.0426</td>
<td><strong>0.0529</strong></td>
<td><strong>0.0529</strong></td>
<td>0.0519</td>
<td>0.0475</td>
<td>0.0473</td>
</tr>
<tr>
<td>189</td>
<td>0.0002</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0004</td>
<td><strong>0.0004</strong></td>
</tr>
<tr>
<td>190</td>
<td><strong>0.0016</strong></td>
<td>0.0012</td>
<td>0.0012</td>
<td>0.0014</td>
<td>0.0013</td>
<td>0.0014</td>
</tr>
<tr>
<td>191</td>
<td><strong>0.0052</strong></td>
<td>0.0022</td>
<td>0.0022</td>
<td>0.0030</td>
<td>0.0012</td>
<td>0.0050</td>
</tr>
<tr>
<td>192</td>
<td><strong>0.0004</strong></td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0003</td>
</tr>
<tr>
<td>193</td>
<td>0.0000</td>
<td>0.0005</td>
<td>0.0005</td>
<td>0.0003</td>
<td>0.0006</td>
<td><strong>0.0010</strong></td>
</tr>
<tr>
<td>194</td>
<td>0.1496</td>
<td>0.1593</td>
<td>0.1594</td>
<td>0.1578</td>
<td><strong>0.1608</strong></td>
<td>0.1541</td>
</tr>
<tr>
<td>195</td>
<td>0.0143</td>
<td>0.1183</td>
<td>0.1183</td>
<td><strong>0.1439</strong></td>
<td>0.1014</td>
<td>0.1353</td>
</tr>
<tr>
<td>196</td>
<td><strong>0.1134</strong></td>
<td>0.0705</td>
<td>0.0705</td>
<td>0.0755</td>
<td>0.0561</td>
<td>0.0971</td>
</tr>
</tbody>
</table>

| MAP      | 0.0348           | 0.0364| 0.0375| 0.0376| 0.0353| **0.0391**|

TRECVID 2006 test runs. This may be largely due to substantial errors introduced by the automatic concept detection. A general observation during TRECVID 2006 was that automatic concept detection with visual features was less effective compared to TRECVID 2005, mostly because over 32% of the TRECVID 2006 collection is footage that is not represented in the development collection [Smeaton and Ianeva, 2006]. The proportion of non-English sources in the 2006 collection is substantially larger than in the 2005 collection, and so speech-based search techniques suffer more from the poorer quality of machine translated text sources. This has a direct impact on the search results, which are on average substantially lower than the results reported at TRECVID 2005 [Smeaton and Ianeva, 2006]. The proportion of rele-
6.2. A FUSION APPROACH TO CONTENT-BASED VIDEO SEARCH

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Run 6 (Baseline)</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>173</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>174</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>175</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0500</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>176</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>177</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
<tr>
<td>178</td>
<td>0.5000</td>
<td>0.5500</td>
<td>0.5000</td>
<td>0.5500</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>179</td>
<td><strong>0.2500</strong></td>
<td>0.1500</td>
<td><strong>0.2500</strong></td>
<td>0.1500</td>
<td>0.1500</td>
<td><strong>0.2500</strong></td>
</tr>
<tr>
<td>180</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>181</td>
<td>0.0000</td>
<td>0.1500</td>
<td>0.0000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0000</td>
</tr>
<tr>
<td>182</td>
<td><strong>0.3000</strong></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>183</td>
<td>0.2000</td>
<td>0.1000</td>
<td>0.1000</td>
<td><strong>0.2500</strong></td>
<td>0.1000</td>
<td>0.2000</td>
</tr>
<tr>
<td>184</td>
<td>0.0500</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
</tr>
<tr>
<td>185</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0000</td>
<td><strong>0.2500</strong></td>
<td>0.0500</td>
</tr>
<tr>
<td>186</td>
<td>0.0000</td>
<td><strong>0.3000</strong></td>
<td>0.3000</td>
<td>0.0000</td>
<td>0.1500</td>
<td>0.1000</td>
</tr>
<tr>
<td>187</td>
<td>0.1500</td>
<td>0.1500</td>
<td>0.1500</td>
<td>0.1500</td>
<td>0.1500</td>
<td><strong>0.1500</strong></td>
</tr>
<tr>
<td>188</td>
<td>0.3500</td>
<td>0.4000</td>
<td>0.4000</td>
<td><strong>0.4500</strong></td>
<td>0.4000</td>
<td>0.4000</td>
</tr>
<tr>
<td>189</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>190</td>
<td>0.0000</td>
<td><strong>0.0500</strong></td>
<td><strong>0.0500</strong></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>191</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>192</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>193</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>194</td>
<td>0.4000</td>
<td>0.3500</td>
<td>0.3500</td>
<td>0.3500</td>
<td><strong>0.4000</strong></td>
<td>0.3500</td>
</tr>
<tr>
<td>195</td>
<td>0.2500</td>
<td>0.4000</td>
<td>0.4000</td>
<td><strong>0.5500</strong></td>
<td>0.3000</td>
<td>0.4500</td>
</tr>
<tr>
<td>196</td>
<td>0.3000</td>
<td><strong>0.5000</strong></td>
<td><strong>0.5000</strong></td>
<td><strong>0.5000</strong></td>
<td><strong>0.5000</strong></td>
<td><strong>0.5000</strong></td>
</tr>
<tr>
<td>Average</td>
<td>0.1188</td>
<td><strong>0.1396</strong></td>
<td>0.1354</td>
<td>0.1354</td>
<td>0.1312</td>
<td>0.1229</td>
</tr>
</tbody>
</table>

Table 6.6: Precision at 20 returned results for each query and for all our submitted runs on the TRECVID 2006 test set. Precision at 20 returned results improves over the text-only baseline in most cases, however, the improvements are not statistically significant.

Relevant shots returned by all participating groups dropped from approximately 18.3% in 2005 to about 9.1% in 2006. The best-performing search run of type B systems at TRECVID 2006 achieved a mean average precision of 0.0753. The lowest performance of the type B systems was at a mean average precision of 0.0114. The average over all type B systems at TRECVID 2006 was at 0.0412. Among our runs, we observe the best mean average precision with Run 5 of 0.0391 — still below the TRECVID average of Type B systems. This run does not use concept term expansion with WordNet, which means that all our term expansion approaches have generally been rather harmful, most likely because of topic drift. Mean Average Precision results of type A runs at TRECVID 2006 were from 0.0005 to 0.0867 with the average at 0.0298. One type C system was tested at TRECVID 2006 that achieved a mean average precision of 0.0022 with its best fully-automatic run.
To better illustrate the retrieval behaviour of our search engine, we plot interpolated recall and precision for our two best runs, Run 1 and Run 5. These are shown in Figure 6.8 along with the results for our baseline run. We observe little variation between these approaches, with small improvements in precision visible at the 10% and 20% recall points.

6.3 Summary

We investigated three complementary automatic query refinement approaches and shown that these have potential for improving speech-based video retrieval. While query expansion performance is query specific, and no single approach emerges as a clear winner across all topics, we observe consistent performance patterns within four manually defined query classes. In particular, each class exhibits different behaviour with respect to the optimal query expansion method. A relatively simple combination approach is able to improve robustness, leading to performance gains for three out of the four query classes, including 23% and 37% gains on the Objects and Setting/Scene classes. Despite weaker results for queries
6.3. SUMMARY

Concerning activities and events, our fusion approach achieves an 11% improvement in mean average precision over all topics.

This work also shows that query-dependent fusion approaches have considerable potential for further improvements. The potential gains demonstrated by the hypothetical Oracle fusion approach range from 8% to 69% for specific query classes, and we observe a 34% potential gain over all queries of the TRECVID 2005 test collection. This shows that we should aim to develop query-dependent methods or weighted fusion. However, query-dependent approaches such as the one presented by Kennedy et al. [2005] require large, independent training sets of topics and ground-truth, which are often not available.

In the second part of this chapter, we tested fully automatic search approaches that incorporate automatically detected concept terms from the visual modality. In variations of these approaches we apply lexical semantic referencing at indexing time to include related terms along with the original terms, and a query-dependent technique to invoke a specialised search based on semantics in the query text. We could not achieve any statistically significant improvements over the speech-only baseline in blind runs on the TRECVID 2006 test set, and the overall results in those runs are generally too weak for practical application. The semantic concept terms that we add to the inverted index do not seem to map well to the visual content. Due to the differences between development and test collections, our hypothesis of approximately similar concept distributions is most likely not valid. In combination with the term expansion using lexical semantic referencing, this leads to severe topic drift in most of our runs and we achieve best performance without term expansion.

The relatively complex query formulations of the TRECVID 2006 test set compound the problem. Often, the 2006 queries combine different types of information needs, such as a person performing a particular activity or in a specific setting. The semantic concept terms that are automatically detected based on visual low-level features of static key frames cannot reflect such complex information well. This highlights the importance of search systems providing sophisticated query pre-processing. For visual search to be effective with complex, free-text queries, comprehensive query analysis and reformulation is required to identify the information need, and apply a specialised retrieval technique.

The results of our training runs on the TRECVID 2005 collection show that the techniques presented here can improve effectiveness of text-based search provided that semantic concept terms reflect the visual content well. Moreover, all our experiments suggest that query refinement and term expansion only work effectively if the parameters can be set properly.
Chapter 7

Conclusions and Future Work

“Solutions almost always come from the direction you least expect, which means there’s no point in trying to look in that direction because it won’t be coming from there.”

– The Salmon of Doubt by Douglas Noël Adams

In this thesis, we developed an effective video shot segmentation algorithm and proposed to use latent class modelling for the unambiguous semantic classification of the video shots. We also established several recommendations about how to best conduct collaborative annotation of visual content, and proposed a combination approach for query refinement in speech-based video search that improves the precision across a wide range of query topics.

We addressed the problem of video shot segmentation by improving a reliable existing technique to detect abrupt transitions, and by combining this technique with a novel method for gradual transition detection. We upgraded the cut detection technique to use localised histograms and replaced a fixed threshold with an adaptive threshold computation. However, the main contribution to research in temporal video segmentation is our novel method for gradual transition detection that examines inter-frame differences within several consecutive frames in video streams.

Moving on from the fundamental issue of temporal video segmentation, we proposed the use of latent class modelling for combining multiple, non-uniform judgements of video shots. By resolving disagreement between the judgements, this method helps to provide high-quality training data to facilitate automatic classification of video shots with supervised machine learning techniques. As it is also important to optimise the annotation process itself, we established several recommendations that help to improve resource utilisation and effectiveness when collecting multiple judgements of visual content. These recommendations
7.1. LESSONS LEARNED

result from the detailed evaluation of two related annotation experiments, and contribute to improving the quality of the training data.

Our target application is content-based video search using free-text and semantic concepts. We explored several query refinement techniques and proposed a combination approach when using speech transcripts for video search to overcome the deficiencies of the individual methods. We also investigated a fusion approach which uses concept terms from automatic video classification in combination with spoken transcripts. In this approach, we applied term expansion using lexical semantic referencing when building the index. While not all our techniques improve the search performance significantly, our findings highlight shortcomings that need to be addressed in the future, and therefore contribute to the advancement of research in content-based video retrieval.

7.1 Lessons Learned

When developing our shot segmentation algorithm, we focused on using colour histogram features that can be extracted efficiently from digital video. We showed that our approach can be highly effective with localised histograms in the HSV colour space. Our technique outperforms most other approaches presented at TRECVID with the added benefit of requiring only one low-level feature to be extracted from the video. While this allows the implementation of efficient shot segmentation systems, it opens avenues for future improvements by using additional features. The results that we observe on the TRECVID shot boundary test collections also show that the detection of gradual transitions remains more difficult than detecting cuts. The detection of very short gradual transitions has been shown to be less effective with our approach.

Due to the popularity of supervised learning algorithms in video retrieval, manual video annotation has become an important part of multimedia research. We showed that latent class modelling helps to model a more generic view of visual content by combining multiple, non-uniform ratings. This way, automatic approaches are enabled to generate a more universal classification instead of only representing an individual opinion. Moreover, modelling multiple ratings with latent class modelling helps to assess annotation quality with the classification performance index $\lambda$. This index is a good indicator of disagreement between user ratings that is not biased by trait prevalence and the skewedness of the concept distribution. The classification performance index is thus a good measure of the confidence that we can have in the annotations for an individual concept. An important benefit of latent
class modelling is that it maximises the number of examples that become available for training. The risk of incorporating false negatives can be minimised by taking the classification performance index into account when selecting examples.

Efficiency and resource utilisation are also important issues when annotating visual content on a larger scale. Our analysis of two collaborative annotation experiments helps to better understand the sources of ambiguity and disagreement between multiple annotators. It is generally beneficial to specify a controlled concept vocabulary for annotation as this enhances consistency and completeness. We did not find any general dependency of annotation time or inter-rater agreement on the type of concept that is annotated, but individual concepts need to be revisited for better specification. This should be done in context with the cultural and educational background of the annotators. We identified the concepts Vegetation, Desert, and Studio as potentially difficult concepts to annotate. Concepts that specify roles, such as Corporate Leader and Police/Security are generally problematic because they require specific domain knowledge. The attempt at annotating and classifying concepts such as Corporate Leader and Government Leader seems questionable because the role that a person holds may change which frequently invalidates such annotations. Our results also show that it is beneficial to annotate with multiple thumbnails per page when using key frames for video shot annotation. Using multiple thumbnails per page preserves temporal context and has not been shown to lead to more disagreement or longer annotation times. Annotating multiple concepts simultaneously appears to be the preferable mode of annotation on the basis of using five concepts. Our analysis shows that using five concepts at a time is faster than using only one concept, while not leading to higher disagreement.

For video search, our score averaging approach to combining several query refinement methods in speech-based retrieval has shown to reduce the topic-dependency of the individual methods. Our overall results during training runs and blind test runs on different collections are improved with this global combination approach. A hypothetical Oracle fusion that is based on a posteriori knowledge of the performance of individual approaches shows that query-dependent fusion has the potential for further improvements. We also experimented with a fusion of speech-text and semantic concept terms extracted from automatic visual classification of video shots. We combined both concept terms and spoken text in one inverted index and applied a lexical expansion technique to the concept terms. While the training results are promising, we could not achieve statistically significant improvements in blind runs on a different test collection. This is mainly because the automatically detected concept terms do not map as well to the test collection as they do to our training collection. Our
combination strategy is based on the assumption that the concept distributions of both development and test collections are similar. This is the case for the 2005 collection that we used for training but not for the 2006 collection which we used to conduct blind runs. In combination with our corpus-independent term expansion technique, this leads to severe topic drift.

The results of these experiments highlight the difficulty of developing a multimodal search approach. Automatic semantic classification is currently helpful for narrow domains where the target collection is relatively similar to the training collection. Query refinement and term expansion techniques require careful parameter tuning which is only possible with suitable training collections. Corpus-independent term expansion, as we used it with WordNet in our experiments, has not been shown to be useful. Our experiments also show that machine translating the speech text corpus introduces a word error rate that substantially reduces the search performance.

### 7.2 Applications and Future Work

As discussed, the obvious application of shot segmentation is in video retrieval, as it helps to generate a manageable and semantically coherent unit for storage and retrieval. Complexity can be reduced using key frames to represent each shot, and video shots may be clustered to form stories in which each shot shares relevance to a given topic. There are other applications for video shot boundary detection: For example, Hoad and Zobel [2003] proposed to use cut detection in digital rights management for video. By analysing the pattern of abrupt transitions in a video stream, their method can be used to identify co-derivatives of video segments in large databases — an application that currently gains importance due to the amount of copyright protected footage being illegally published on the Internet [La Monica, 2007; Sandoval, 2007]. While Hoad and Zobel [2003] only considered abrupt transitions in their work, it would be interesting to use gradual transitions as well as abrupt transitions since this could enhance detection performance.

Future work should continue to improve the gradual transition detection performance as it still falls short of the performance in cut detection. Our technique of using the average frame similarity throughout several successive frames is a good basis of further research in this area. Other low-level features could be incorporated to investigate whether robustness and effectiveness can be improved. Our method could also be extended to monitor two Pre-Post Ratio curves, one for the full window and one for a smaller part of the window. While
this would increase the computational complexity, it could help to improve effectiveness when detecting very short gradual transitions.

Latent class modelling is used in the social, behavioural, and medical sciences to model survey responses and data from medical diagnostics. It has also been used in information retrieval for indexing text-document collections [Hofmann, 1999]. We have shown its usefulness for video content modelling, but an outstanding question is whether the performance of supervised classification algorithms can be improved by using training data generated with latent class modelling. Another question is whether the posterior class membership probabilities and the classification performance index could be incorporated into the training process. This could, for example, be done in the form of a confidence score for each example. Future work should also include an exploration of alternative models. While we used a restricted two-class model, other models with more classes might be useful to better capture the ambiguity of visual content. As described in Chapter 4, due to the nature of our data, we are required to apply a restriction to our latent class model to cater for rater interchangeability. However, this does not reflect reality well, as it assumes that all human reviewers are equal. We should aim at removing this restriction; this may be achieved by planning annotation tasks in a way that permits the collection of adequate data for such a modelling process.

In light of popular community tagging applications on the Internet in which users assign textual labels to images and video, it is also interesting to address the question whether we can use these annotations within an unsupervised learning framework such as latent class modelling. We also see applications for latent class modelling as an intelligent fusion technique to combine search results from multiple modalities. This way, we could model the response behaviour of different search approaches, and enable query-dependent, weighted fusion. More generally, latent class modelling can be used as an approach to any kind of sensor fusion where several input sources need to be combined. For example, in security and safety applications using multiple cameras from different angles or operating at different wavelengths, latent class modelling would allow us to combine the input sources and compute a probability that a given event has occurred. This requires, however, that the input sources’ response behaviour can be modelled over a larger number of samples.

We were able to draw useful conclusions from analysing two experiments concerned with collaborative manual annotation of visual content. Our recommendations can improve annotation quality in collaborative annotation efforts, and help to maximise resource utilisation. However, research in this area is still at its early stages and more large-scale user-studies are needed to enhance the understanding of the sources for human error and inter-rater disagree-
7.2. APPLICATIONS AND FUTURE WORK

While we showed that annotating up to five semantic concepts simultaneously does not lead to increased disagreement between users, we cannot be sure of the effects of annotating more than five concepts. Moreover, in our work, we did not address issues regarding interface design and only considered categorical annotation using up to three categories. As outlined above, latent class modelling does allow modelling multiple-class categorical responses and even continuous responses. Future research should address questions about how this could be translated into an annotation task and how we could use such annotation to train supervised learning algorithms.

Our results in video search show that we should aim to develop query-dependent approaches. Many individual search techniques exist, each with their own merits for answering particular information needs, but we need to provide more generic solutions. Kennedy et al. [2005] showed that it is useful to be able to automatically discover query classes, and answer each type of query with an optimised approach. Research in this area is hampered by the limited availability of realistic video search training queries. While the TRECVID test collections and queries are generally useful, even all collections combined do not yield enough queries for effective query-class mining. Another aspect that has not yet been fully addressed is the problem of multilingual retrieval when using speech text for video search. The frequently used approach of machine translating the corpus does not seem to be suitable when the translation process adds many word errors. Instead, translating and disambiguating the query, and performing the search in the native language could improve results, as suggested by the work of Zhang and Vines [2004] in cross-lingual text-document retrieval.

The work presented in this thesis highlights the diversity of research in video retrieval. Our methods help to perform effective temporal segmentation, classification, and search of video content, although our results also show that content-based video retrieval remains challenging. Besides providing solutions to some of the problems in video retrieval, our work opens interesting new lines for future research in multimedia content modelling and retrieval.
Appendix A

TRECVID Shot Boundary Test Collections (2001 – 2006)

<table>
<thead>
<tr>
<th>TRECVID 2001:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clips:</td>
</tr>
<tr>
<td>Total duration:</td>
</tr>
<tr>
<td>Collection size:</td>
</tr>
<tr>
<td>Description:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition statistics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
</tr>
<tr>
<td>Dissolve</td>
</tr>
<tr>
<td>Fade</td>
</tr>
<tr>
<td>Other</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Short graduals</td>
</tr>
</tbody>
</table>

\textsuperscript{1}http://www.open-video.org
### TRECVID 2002:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clips</td>
<td>18 (average duration: 16min. 51sec.)</td>
</tr>
<tr>
<td>Total duration</td>
<td>5hrs. 3min. 13sec.</td>
</tr>
<tr>
<td>Collection size</td>
<td>~ 2.7 GB</td>
</tr>
<tr>
<td>Description</td>
<td>Television documentaries and interview programme from the 1960s, 1970s, and 1980s, digitised from analogue recordings.</td>
</tr>
</tbody>
</table>

#### Transition statistics:

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
<th>Percentage</th>
<th>Average transition length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
<td>1466</td>
<td>70.1%</td>
<td>29.4 frames</td>
</tr>
<tr>
<td>Dissolve</td>
<td>511</td>
<td>24.4%</td>
<td>29.4 frames</td>
</tr>
<tr>
<td>Fade</td>
<td>63</td>
<td>3.0%</td>
<td>87.9 frames</td>
</tr>
<tr>
<td>Other</td>
<td>50</td>
<td>2.4%</td>
<td>16.8 frames</td>
</tr>
<tr>
<td>Total</td>
<td>2090</td>
<td></td>
<td>34.3 frames</td>
</tr>
</tbody>
</table>

Short graduals: 33 (5.3% of all gradual transitions)

### TRECVID 2003:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clips</td>
<td>13 (average duration: 26min. 40sec.)</td>
</tr>
<tr>
<td>Total duration</td>
<td>5hrs. 46min. 41sec.</td>
</tr>
<tr>
<td>Collection size</td>
<td>~ 4.6 GB</td>
</tr>
<tr>
<td>Description</td>
<td>Television news and parliament debates from 1998-2001, recorded from U.S. American channels such as CNN, ABC, and C-SPAN. The news are interrupted by advertisement and short entertainment sections.</td>
</tr>
</tbody>
</table>

#### Transition statistics:

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
<th>Percentage</th>
<th>Average transition length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
<td>2644</td>
<td>70.8%</td>
<td>15.9 frames</td>
</tr>
<tr>
<td>Dissolve</td>
<td>753</td>
<td>20.2%</td>
<td>15.9 frames</td>
</tr>
<tr>
<td>Fade</td>
<td>116</td>
<td>3.1%</td>
<td>29.2 frames</td>
</tr>
<tr>
<td>Other</td>
<td>221</td>
<td>5.9%</td>
<td>25.7 frames</td>
</tr>
<tr>
<td>Total</td>
<td>3734</td>
<td></td>
<td>19.3 frames</td>
</tr>
</tbody>
</table>

Short graduals: 78 (7.2% of all gradual transitions)
### TRECVID 2004:

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clips</td>
<td>12 (average duration: 28min. 40sec.)</td>
</tr>
<tr>
<td>Total duration</td>
<td>5hrs. 43min. 57sec.</td>
</tr>
<tr>
<td>Collection size</td>
<td>~ 4.3 GB</td>
</tr>
<tr>
<td>Description</td>
<td>Television news from 1998, recorded from U.S. American channels such as CNN and ABC. The news are interrupted by advertisement and short entertainment sections.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
<td>2776 (57.7%)</td>
</tr>
<tr>
<td>Dissolve</td>
<td>1525 (31.7%) Average transition length: 13.6 frames</td>
</tr>
<tr>
<td>Fade</td>
<td>230 (4.8%) Average transition length: 30.5 frames</td>
</tr>
<tr>
<td>Other</td>
<td>276 (5.7%) Average transition length: 22.8 frames</td>
</tr>
<tr>
<td>Total</td>
<td>4807 Average transition length: 16.8 frames</td>
</tr>
<tr>
<td>Short graduals</td>
<td>486 (23.9% of all gradual transitions)</td>
</tr>
</tbody>
</table>

### TRECVID 2005:

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clips</td>
<td>13 (average duration: 36min. 21sec.)</td>
</tr>
<tr>
<td>Total duration</td>
<td>7hrs. 52min. 29sec.</td>
</tr>
<tr>
<td>Collection size</td>
<td>~ 4.8 GB</td>
</tr>
<tr>
<td>Description</td>
<td>Television news from U.S. American, Chinese, and Arabic channels from 2004, interspersed by advertisement and short entertainment segments. In addition, this collection contains four promotional videos from NASA.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
<td>2759 (60.8%)</td>
</tr>
<tr>
<td>Dissolve</td>
<td>1382 (30.5%) Average transition length: 12.3 frames</td>
</tr>
<tr>
<td>Fade</td>
<td>81 (1.8%) Average transition length: 33.0 frames</td>
</tr>
<tr>
<td>Other</td>
<td>313 (6.9%) Average transition length: 20.6 frames</td>
</tr>
<tr>
<td>Total</td>
<td>4535 Average transition length: 14.7 frames</td>
</tr>
<tr>
<td>Short graduals</td>
<td>621 (35.0% of all gradual transitions)</td>
</tr>
<tr>
<td>TRECVID 2006:</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Number of clips: 13 (average duration: 25min. 33sec.)</td>
<td></td>
</tr>
<tr>
<td>Total duration: 5hrs. 32min. 4sec.</td>
<td></td>
</tr>
<tr>
<td>Collection size: (~ 4.3 \text{ GB})</td>
<td></td>
</tr>
<tr>
<td>Description: Television news from U.S. American, Chinese, and Arabic channels from 2005. The news</td>
<td></td>
</tr>
<tr>
<td>sections are interrupted by advertisement and short entertainment segments.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition statistics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuts</td>
</tr>
<tr>
<td>Dissolve 1509 (39.9%) Average transition length: 8.5 frames</td>
</tr>
<tr>
<td>Fade 51 (1.3%) Average transition length: 24.0 frames</td>
</tr>
<tr>
<td>Other 381 (10.1%) Average transition length: 16.3 frames</td>
</tr>
<tr>
<td>Total 3785 Average transition length: 10.4 frames</td>
</tr>
<tr>
<td>Short graduals 915 (47.1% of all gradual transitions)</td>
</tr>
</tbody>
</table>
Appendix B

TRECVID 2005 Annotation Forum Concepts

The 44 semantic concepts as they were defined by Naphade et al. [2005] for TRECVID 2005. 39 of these concepts were used for the manual annotation of the TRECVID 2005 development corpus during the annotation forum. The five concepts marked with (*) were not manually annotated.

A 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 advertisements, commercials

B Setting/Scene/Site

1. Indoor*: Shots of Indoor locations
2. Court: Shots of the interior of a court-room location
3. Office: Shots of the interior of an Office Setting
4. Meeting: Shots of a Meeting taking place indoors
5. Studio Setting: Shots of the studio setting including anchors, interviews and all events that happen in a news room
6. Outdoor: Shots of Outdoor locations
7. Building: Shots of an exterior of a building
8. Desert: Shots with the desert in the background
9. Vegetation: Shots depicting natural or artificial greenery, vegetation woods, etc.
10. Mountain: Shots depicting a mountain or mountain range with the slopes visible
11. Road: Shots depicting a road
12. Sky: Shots depicting sky
13. Snow: Shots depicting snow
14. Urban-Setting: Shots depicting an urban or suburban setting
15. Waterscape/Waterfront: Shots depicting a waterscape or waterfront

C People
1. Crowd: Shots depicting a crowd
2. Face: Shots depicting a face
3. Person: Shots depicting a person. The face may or may not be visible
4. Government Leader: Shots of a person who is a governing leader e.g. president, prime-minister, chancellor of the exchequer, etc.
5. Corporate Leader: Shots of a person who is a corporate leader e.g. CEO, CFO, Managing Director, Media Manager etc.
6. Police/Private Security Personnel: Shots depicting law enforcement or private security agency personnel
7. Military: Shots depicting the military personnel
8. Prisoner: Shots depicting a captive person, e.g., imprisoned, behind bars, in jail, in handcuffs, etc.

D Objects
2. Computer or Television Screens: Shots depicting television or computer screens
3. Flag-US: Shots depicting a US flag
4. Airplane: Shots of an airplane
5. Car: Shots of a car
6. Bus: Shots of a bus
7. Truck: Shots of a truck
8. Boat/Ship: Shots of a boat or ship

E Activities
1. Walking/Running: Shots depicting a person walking or running
2. People Marching: Shots depicting many people marching as in a parade or a protest

F Events
1. Explosion/Fire: Shots of an explosion or a fire
2. Natural Disaster: Shots depicting the happening or aftermath of a natural disaster such as earthquake, flood, hurricane, tornado, tsunami

G Graphics
1. Maps: Shots depicting regional territory graphically as a geographical or political map
2. Charts: Shots depicting any graphics that is artificially generated such as bar graphs, line charts etc. Maps should not be included
Appendix C

Selected TRECVID Search Corpora

The TRECVID search collections for each year are divided into two approximately equal-sized sets, one for system development and one for blind test runs. In addition to each year’s development collection, all TRECVID test and development collections from previous years may be used for system development and training. Both development and test sets consist of the video files, speech transcripts, and a shot segmentation reference, including one representative frame for each shot. Both shot segmentation and key frame extraction are generated in an automated process. This means that the reference may be imperfect, but the focus is on providing a common reference to keep results comparable and to enable participation of groups who are not able to perform shot segmentation. The speech transcripts are generated by Automatic Speech Recognition (ASR) or originate from Closed Captions (CC) if available.

The search test queries — usually 24 — to perform blind runs are released only shortly before the submissions are due. The submissions in form of a ranked result list with up to 1000 results per query are manually evaluated at NIST. This evaluation is released after the workshop as a pooled ground-truth so that it can be used for development and training in the future. The development collections usually remain the same for two years, only a new test collection and queries are provided each year. We provide statistics of the TRECVID search collections from 2003, 2005, and 2006 below:
### TRECVID 2003 test set:

<table>
<thead>
<tr>
<th>Number of clips</th>
<th>113</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shots</td>
<td>32,318</td>
</tr>
<tr>
<td>Total duration</td>
<td>(\sim 63) hours</td>
</tr>
<tr>
<td>Collection size</td>
<td>(\sim 50) GB</td>
</tr>
</tbody>
</table>

### TRECVID 2003 development set:

<table>
<thead>
<tr>
<th>Number of clips</th>
<th>133</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shots</td>
<td>35,067</td>
</tr>
<tr>
<td>Total duration</td>
<td>(\sim 63) hours</td>
</tr>
<tr>
<td>Collection size</td>
<td>(\sim 50) GB</td>
</tr>
</tbody>
</table>

**Description:** Both test and development set feature television news from U.S. English sources (ABC, CNN, C-SPAN) from the years 1998 to 2001, interspersed with advertisement sections.

### TRECVID 2005 test set:

<table>
<thead>
<tr>
<th>Number of clips</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shots</td>
<td>45,765</td>
</tr>
<tr>
<td>Total duration</td>
<td>(\sim 81) hours</td>
</tr>
<tr>
<td>Collection size</td>
<td>(\sim 62) GB</td>
</tr>
</tbody>
</table>

### TRECVID 2005 development set:

<table>
<thead>
<tr>
<th>Number of clips</th>
<th>137</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shots</td>
<td>61,904</td>
</tr>
<tr>
<td>Total duration</td>
<td>(\sim 81) hours</td>
</tr>
<tr>
<td>Collection size</td>
<td>(\sim 62) GB</td>
</tr>
</tbody>
</table>

**Description:** Both test and development set feature television news from U.S. English (NBC, CNN, MSNBC), Chinese (CCTV4, NTDTV), and Arabic (LBC) channels from 2004. The news sections are interrupted by advertisement and short entertainment segments. 44% of the test collection are in U.S. English, 31% in Chinese, and 25% in Arabic.
TRECVID 2006 test set:

<table>
<thead>
<tr>
<th>Number of clips:</th>
<th>259</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shots:</td>
<td>79,484</td>
</tr>
<tr>
<td>Total duration:</td>
<td>~159 hours</td>
</tr>
<tr>
<td>Collection size:</td>
<td>~127 GB</td>
</tr>
</tbody>
</table>

TRECVID 2006 development set: re-used from 2005

| Description: | Both test and development set feature television news from U.S. English (NBC, CNN, MSNBC), Chinese (CCTV4, PHOENIX, NTDTV), and Arabic (LBC, HURRA) channels from 2005. The news sections are interrupted by advertisement and short entertainment segments. 32% of the 2006 test collection is recordings from television channels that are not represented in any development collection from before. 29% of the test collection are in U.S. English, 19% in Chinese, and 52% in Arabic. |

Acknowledgements

Different TRECVID participants contributed to generate a common shot segmentation reference for these collections [Petersohn, 2004; Quénot et al., 2003], and to extract representative frames for each shot [Cooke et al., 2004; Quénot et al., 2003]. The Informedia Team\(^1\) at Carnegie-Mellon University [Vogel et al., 2003], the Linguistic Data Consortium\(^2\) at the University of Pennsylvania, and the Laboratoire d’Informatique pour la Mécánique et les Sciences de l’Ingénieur (LIMSI) [Gauvain et al., 2002] contributed speech transcripts leveraging automatic speech recognition and closed captions where available. In 2005 and 2006, the non-English sources were also machine translated using state-of-the-art software such as the Virage VideoLogger\(^3\), Language Weaver\(^4\), and the BBN Byblos\(^5\) system. More information can be found in the overview papers of the respective TRECVID workshops [Kraaij et al., 2006; Over et al., 2005; Smeaton et al., 2003].

\(^{1}\)http://www.informedia.cs.cmu.edu  
\(^{2}\)http://www.ldc.upenn.edu  
\(^{3}\)http://www.virage.com/content/products/index.en.html  
\(^{4}\)http://www.languageweaver.com  
\(^{5}\)http://www.bbn.com
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


175
BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


D. Metzler and W. B. Croft. Combining the Language Model and Inference Network Approaches to Retrieval. Information Processing and Management Special Issue on Bayesian


J. Nang, S. Hong, and Y. Ihm. An Efficient Video Segmentation Scheme for MPEG Video Stream using Macroblock Information. In J. Buford, S. Stevens, D. Bulterman, K. Jeffay,


BIBLIOGRAPHY


196


BIBLIOGRAPHY


