FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data Organization and Search in Multimedia Databases

Catalin Mihai Calistru

Programa Doutoral em Engenharia Informática

Supervisor: Cristina Ribeiro

Co-Supervisor: Gabriel David

September, 2008

Abstract

The wealth of multimedia items and their increasing complexity make data organization and search essential. Without efficient storage, accurate search and informative retrieval it is hard to explore a multimedia repository to its full potential.

The multimedia items present strong connections between the content data and their metadata. The content can offer by itself, or upon automatic content analysis, important low-level information about colors, shapes or sounds which is called content metadata. But there are more metadata, that are unlikely to be automatically extracted from content which are equally important. The common set called contextual metadata includes title, author, date, origin details or annotations.

It is a fact that both content and metadata are essential in multimedia information retrieval. There are plenty of standardization efforts that embed metadata in text wrappers in order to help on its processing throughout the multimedia item life cycle.

By far the most important problem in multimedia retrieval is the so-called “semantic gap”. It expresses the lack of a direct semantic channel between object features such as color, texture or shape and the concepts that one has in mind when formulating a query. Besides the “semantic gap”, the multimedia retrieval systems must also tackle the problems that come from the heterogeneity of the metadata standards and the nature of the datatypes that they introduce. Rising from different multimedia communities and embodying different perspectives, the metadata standards define sets of concepts for their domains of activity. However, it is not easy to find mappings between the sets of concepts, even in the cases where overlapping exists. Indexing a wide range of metadata datatypes is another challenge. The vectorial datatypes, although embedded in text, are numeric in nature. For instance, the Scalable Color Descriptor defined by the MPEG-7 standard, is a vector of 256 values that requires high-dimensional indexing methods.

We argue that the problems of managing large and heterogeneous multimedia repositories can be alleviated by bringing improvements in several aspects: storage model, high-dimensional indexing, and retrieval. Three main contributions are proposed: a database model, a high-dimensional indexing method and a faceted retrieval system.

The proposed database model accounts for content organization and its association to metadata. It is a hybrid relational-XML database model. The model allows content segments and subsegments to be arranged in configurable hierarchical structures. The associations between content and metadata are based on a set of concepts from archival description and multimedia standards. Context metadata together with the structure of the multimedia items are stored in the relational part of the model. The metadata descriptors that contain high dimensional data are stored into the XML part of the model.

For search within the high-dimensional descriptors, an indexing method called BitMatrix is proposed. It constructs bit signatures that can be efficiently processed with bitwise operations. Our experiments have shown that the use of the BitMatrix as a high-dimensional indexing method is beneficial for the retrieval process.

Finally, the MetaMedia retrieval system was developed. Built as a web application, MetaMedia has a user-interface (the client) and a server that hosts the retrieval system itself; the proposed database model and the BitMatrix index are instantiated in MetaMedia.

In a first set of experiments, MetaMedia has been implemented in two case studies. The first one is a historic documentation center of "Santa Maria da Feira" that allows the visualization and search of its documents based on their textual content and contextual metadata. The other one is "Enthrone", a multimedia distribution framework that has used MetaMedia as a multimedia repository. The main functionalities have been storing and searching multimedia items based on contextual metadata.

MetaMedia has also been evaluated in TRECVID, a well-known video retrieval benchmark, as an automatic and interactive video retrieval system that combined content features such as color, texture, shape and audio features with annotations. The queries included natural language, image and video components. The comparative results illustrate satisfactory performance. Separate evaluations of the BitMatrix index were performed in a custom-designed multidimensional indexing evaluation framework.

Resumo

A riqueza dos itens multimédia e a sua crescente complexidade torna essenciais operações de organização e pesquisa de dados. Sem armazenamento eficiente, precisão nas pesquisas e recuperação informativa é difícil explorar um repositório multimédia no seu pleno potencial.

Os itens multimédia apresentam ligações fortes entre conteúdo e meta-informação. O conteúdo pode oferecer, por si próprio ou mediante análise automática, descritores de baixo nível sobre cores, formas, ou sons, que são chamados meta-informação de conteúdo. Mas há outra meta-informação, que dificilmente é extraída automaticamente a partir do conteúdo, e que é igualmente importante. A chamada meta-informação de contexto inclui habitualmente título, autor, data, detalhes de origem ou anotações.

É um facto que tanto os conteúdos como a meta-informação são essenciais em recuperação de informação multimédia. Vários esforços de normalização propõem formatos textuais para a meta-informação, a fim de ajudar o processamento desta ao longo de todo o ciclo de vida dos itens multimédia.

O maior problema na recuperação multimédia é o chamado "fosso semântico" que se revela na falta de um canal semântico directo entre propriedades dos objetos como côr, textura ou forma e conceitos que um utilizador qualquer tem em mente ao formular uma pergunta. Para além do "fosso semântico", os sistemas de recuperação multimédia têm também de resolver os problemas que provêm da heterogeneidade das normas de meta-informação e da natureza dos tipos de dados que estas introduzem. Tendo origem em diversas comunidades com diferentes perspectivas, as normas definem conjuntos de conceitos para os seus domínios de actividade. No entanto, não é fácil encontrar mapeamentos entre os conceitos, mesmo nos casos em que existem sobreposições. Um outro desafio vem da vasta gama de tipos de meta-informação a indexar. Os tipos vectoriais, embora embebidos no texto, são de natureza numérica. O "Scalable Color Descriptor", por exemplo, definido pela norma MPEG-7, é um vector de 256 valores que exige métodos de indexação multidimensional.

Defendemos que os problemas da gestão de repositórios multimédia grandes e heterogéneos podem ser minorados com melhorias em diversos aspectos: o modelo de armazenamento, a indexação multidimensional e a recuperação de informação. Propõem-se três contribuições principais: um modelo de dados, um método de indexação multidimensional e um sistema de recuperação facetada.

O modelo de dados contribui para a organização dos conteúdos e a sua associação à meta-informação. O modelo é híbrido, tendo uma parte relacional e outra XML e permite representar partes do conteúdo, chamadas segmentos, em estruturas hierárquicas configuráveis. As associações entre conteúdo e meta-informação são baseadas num conjunto de conceitos usados nas normas arquivísticas e de multimédia. A meta-informação contextual juntamente com a estrutura dos itens multimédia são armazenados na parte relacional do modelo. A meta-informação de conteúdo, normalmente descritores de natureza multidimensional, são armazenados na parte XML do modelo.

Para a pesquisa nos descritores multidimensionais, propomos um método de indexação designado BitMatrix. O método baseia-se na construção de assinaturas de bits que podem ser eficientemente processadas com operações ao bit. As experiências realizadas mostram que o uso do BitMatrix como método de indexação multidimensional é benéfico para o processo de recuperação.

A terceira contribuição é o sistema de recuperação MetaMedia. Desenvolvido como uma aplicação web, o MetaMedia tem uma interface de utilizador (o cliente) e um servidor que integra o sistema de recuperação em si; o modelo de dados proposto e o índice BitMatrix são instanciados no MetaMedia.

Num primeiro conjunto de experiências, MetaMedia foi implementado em dois casos de estudos. O primeiro é um centro de documentação histórica —“ Santa Maria da Feira”— que permite a pesquisa e a visualização dos seus documentos com base nos conteúdos e na meta-informação contextual. O outro é o “ Enthrone”, uma plataforma de distribuição multimédia que usou o MetaMedia como repositório multimédia. As principais funcionalidades são armazenar e pesquisar itens com base na meta-informação contextual.

O MetaMedia foi também avaliado no TRECVID, um “benchmark"internacional de recuperação vídeo, onde funcionou como sistema automático e interativo combinando características de conteúdo tais como cor, textura, forma e áudio com anotações. As interrogações incluiram componentes de linguagem natural, imagem e vídeo. Os resultados comparativos ilustram um desempenho satisfatório. Foram também realizadas avaliações do índice BitMatrix num ambiente criado especialmente para o efeito.

Résumé

La richesse des données multimédia et leur complexité croissante rend l'organisation des données et la recherche essentiels. Sans efficacité de stockage, de recherche précise et informative, il est difficile d'explorer totalement le potentiel de basées de données multimédia.

Les données multimédia présentent des rapports étroits entre le contenu des données et leurs métadonnées. Le contenu peut offrir, par lui-même ou à partir d'une analyse automatique, des descripteurs de bas niveau de couleurs, formes, ou sons, qui sont appelés métadonnées de contenu. Mais il y a d'autres métadonnées peu susceptibles d'être extraites automatiquement d'un contenu, mais qui sont tout aussi importantes. Elles sont appelées métadonnées contextuelles et comprennent le titre, l'auteur, la date, les détails d'origine ou les annotations.

Il est un fait que le contenu et les métadonnées sont essentielles dans la recherche d'information multimédia. Il y a beaucoup d'efforts de normalisation qui intègrent les métadonnées dans des formats textuels, afin d'aider à leur traitement tout au long du cycle de vie des données multimédia.

De loin le problème le plus important dans la recherche multimédia est ce qu'on appelle l' "écart sémantique". Il exprime l'absence d'un canal sémantique direct entre les caractéristiques d'objets telles que la couleur, la texture ou la forme, et les concepts que chaque utilisateur a à l'esprit lors de la formulation d'une question. Outre l' "écart sémantique", les systèmes de recherche multimédia doivent aussi résoudre les problèmes qui proviennent de l'hétérogénéité des normes des métadonnées et la nature des types de données qu'ils fournissent. Provennant de différentes communautés avec des perspectives différentes, les normes définissent des ensembles de concepts pour leurs domaines d'activités. Toutefois, il n'est pas facile de trouver des correspondances entre les ensembles de concepts, même quand un certain chevauchement existe.

Un autre défi vient du large éventail de types de métadonnées à indexer. Les types vecteur, bien que intégrés dans le texte, sont de nature numérique. Par exemple, le "Scalable Color Descriptor" défini par la norme MPEG-7, est un vecteur de 256 valeurs qui nécessite de méthodes d'indexation multidimensionnelles.

Nous soutenons que les problèmes de la gestion des basées de données multimédia grands et hétérogènes peuvent être atténués par des améliorations dans plusieurs aspects : le modèle de stockage, l'indexation multidimensionnelle et la recherche d'information. Trois principales contributions sont proposées : un modèle de données, une méthode d'indexation et un système de recherche multi-facettes.

Le modèle de données contribue à l'organisation du contenu et de son association avec des métadonnées. Le modèle est un hybride, relationnel et XML, qui permet aux parties du contenu appellées segments d'être organisées en structures hiérarchiques configurables. Les associations entre le contenu et les métadonnées sont basées sur un ensemble de concepts utilisés dans les normes arquivistes et du multimédia. Les métadonnées contextuelles, ainsi que la structure des éléments multimédia sont stockés dans la partie relationnelle du modèle. Les métadonnées de contenu, généralement les descripteurs de nature multidimensionnelle, sont stockées dans la partie XML du modèle.

Pour rechercher les descripteurs multidimensionnels, nous proposons une méthode d'indexation appelée BitMatrix. La méthode est basée sur la construction de signatures de bits qui peuvent être traitées efficacement avec les opérations binaires. Les expériences montrent que l'utilisation de BitMatrix comme une méthode d'indexation multidimensionnelle est bénéfique pour le processus de récupération.

La troisième contribution est le système de récupération MetaMedia. Développé comme une application Web, le MetaMedia a une interface utilisateur (le client) et un serveur qui intègre le système de récupération en soi ; le modèle de données proposé et l'indice BitMatrix sont instanciés dans MetaMedia.

Dans une première série d'expériences, MetaMedia a été mis en place dans deux études de cas. Le premier est un centre de documentation historique, "Santa Maria da Feira", permettant de chercher et de consulter les documents en fonction de leur contenu et leurs métadonnées contextuelles. L'autre est " ENTHRONE", une plate-forme de distribution multimédia qui a utilisé le MetaMedia comme référentiel. Les principales caractéristiques ont été le stockage et la recherche basée sur les métadonnées contextuelles.

Le MetaMedia a également été évalué dans TRECVID, un "benchmark" international en vidéo, ou il a fonctionné comme un système automatique et interactif, combinant les éléments de contenus tels que la couleur, la texture, la forme et audio, avec des annotations. Les questions ont été composées du langage naturel, l'image et la vidéo. Les résultats comparatifs montrent une performance satisfaisante. Des évaluations de l'indice BitMatrix dans un environnement créé spécialement à cet effet ont également été faites.

Acknowledgments

Writing the acknowledgments comes to me with contradictory feelings. The relief that the thesis project has been finished is counterbalanced by uncertainty, what will I do next? I just hope I will make a wise decision.

The first research project (MOUMIR) that I have participated in, in late 2002, provided me the contact with the field of multimedia databases. Besides the challenging objective of enabling search operations in a set of RTP videos, this project allowed me to get to know the telecommunication and multimedia group at INESC. Based on discussions and fruitful collaborative work with professors Luís Corte-Real, Cristina Ribeiro and Gabriel David, the idea and the initial work plan for my PhD thesis came up.

I am especially indebted to Cristina Ribeiro and Gabriel David who accepted to be my supervisors. These words are perhaps contained by every acknowledgments chapter, but I mean them. In spite of their busy schedules they were always willing to help and transform the time spent in pleasant moments. With their help during the years, I have been in the fortunate position to successfully combine parts of my PhD work with interesting research projects such as ENTHRONE and DOMIR.

There are several INESC colleagues with whom I cooperated in an intense way during the last four and a half years and who indeed deserve my gratitude: Bruno Gonçalves, Nuno Cerqueira, André Barbosa, Jaime dos Santos Cardoso, Lucian Ciobanu, Georgiana Ciobanu, Luís Gustavo Martins, Luís Filipe Teixeira, Daniel Oancea and Hélder Castro.

Another THANK YOU goes to my young, but already extended family. Weekends stimuli such as “Cata, shouldn’t you be writing on your thesis instead of gardening?” were indeed motivating for completing the manuscript. But the boost came from the two youngest members. Totaling less than 30 months of life, they have asked for a copy of the manuscript as soon as possible...

Last, but not least, thanks to the One to whom the ultimate semantic levels belong.

“In the beginning was the Verb”

Contents

List of Figures

List of Tables

Acronyms

AP average precision

CBIR Content-based Information Retrieval

DU Description Unit

DIA Digital Item Adaptation

DID Digital Item Declaration

DII Digital Item Identification

DIP Digital Item Processing

IID independent identical distributed

IPMP Intellectual Property Management and Protection

IR Information Retrieval

HSV hue, saturation, value

LM Language-based Models

LSI Latent Semantic Indexing

MAM Metric Access Methods

MAP mean average precision

MIR Multimedia Information Retrieval

MMDI Multidimensional Multimedia Descriptor Indexing

MPEG Moving Picture Experts Group

OCR Optical Character Recognition

PCA Principal Component Analysis

PM Probabilistic Models

PRP Probability Ranking Principle

RDBMS Relational Database Management Systems

REL Rights Expression Language

SAM Spatial Access Methods

SQL Structured Query Language

SVD Singular Value Decomposition

VSM Vector Space Models

XML Extensible Markup Language

Chapter 1

Introduction

Multimedia data is currently generated by all sorts of devices. Meaningful chunks of data such as pictures, videos or sounds can be designated as multimedia items, and the ability to manage them is essential for people and organizations. In areas such as news broadcast or entertainment, multimedia items are the product itself, but in most areas of activity the processes within organizations give rise to information which incorporates complex components.

1.1 Context

Multimedia repositories display a large diversity both in the nature of stored items and in the application domains. They may include:

- objects which have been collected and appropriately described, as in a digital library;
- heterogeneous information assembled and dynamically modified in a Web site;
- the result of the production process of a publisher or a broadcaster.

Repository managers and users need to easily incorporate new items, to make them available in well-established formats and, the most important, to search them [Tes04]. Research in Multimedia Information Retrieval (MIR) has produced increasingly sophisticated search methods [DJLW08, LSDJ06, SW05, Bim99, YI99, GJ97], with techniques that range from the exploration of content features in specific domains [FSN$^+$95, ORC$^+$97, SC96] to the migration of techniques developed for text to image or video [ZG02].

Multimedia items are intrinsically complex, and the analysis of their content uses heavy computing processes. Excepting the case of text, multimedia content analysis do not result in the high-level concepts required by the generality of the search tasks. This brings about the so-called "semantic gap", clearly identified as the main issue in multimedia retrieval [GR95]. Text is a form of communication based on concepts, expressed in the user's language and close to the way he thinks. Searching text items may require a number of processing techniques like pattern matching, stemming, finding synonyms, translating, natural language analysis. Supposing that the user expresses his information need through words, the set of retrieved documents includes those containing exact word matches and can be continuously enlarged to more and more semantically related documents. Although a "picture is worth a thousand words" a text-like semantic channel between a pictures repository and the information need does not exist. The automatic analysis of a picture may produce many descriptors related to the contents, also called low-level descriptors, but hardly produces an accurate descriptive sentence close enough to the query words. Of course, one way to reduce the semantic gap when searching image repositories is to start with a picture example and establish a similarity measure between its automatically obtained descriptors and the ones corresponding to the images in the repository. But this is not a true solution because most of the times the users think and express themselves in concepts and words.

The multimedia items present strong connections between the content data and their metadata. The content can offer by itself, or upon automatic content analysis, important information about text statistics of word usage, automatic speech recognition, colors, shapes, sounds, which is called content metadata. But there is a lot more that is unlikely to be automatically extracted from content such as the title, author, date, origin details or annotations. Such information is generally called context metadata. It is debatable whether or not some metadata can be considered content or context-related and it may to some extent depend on the application domain. For instance, in domains such as broadcasting or video on demand, the title, author or date are considered closer to content while in the archival domain the same attributes are clearly considered "context metadata". However, we follow the criterion presented above: if metadata can be extracted automatically from content it is considered content-related, otherwise it is contextual. Depending on the domain activity of a multimedia community, metadata can also enclose terms of use, copyrights, unique identifiers, technical details such as format or size, adaptation and re-purposing instructions.

Content and context metadata are both essential. If content metadata such as color and shape may completely satisfy a designer, an archivist wishing to certify the authenticity of a document may consider the context metadata as important as the content itself. Many existing and emerging metadata standards are relevant for managing multimedia information, both for the objects themselves and for the corresponding metadata. The standardization efforts define *descriptors* for embedding metadata in text or Extensible Markup Language (XML) wrappers in order to help on its creation, exchange and automatic processing throughout the multimedia life cycle [DE06, KBD$^+$05, Kos03]. It has however been noted that the engagement in standards development might be hindering the efforts to develop operational multimedia retrieval systems [Bul04]. The metadata standards development has contributed to clarify concepts and to promote interoperability, but their goal is not to provide models for the representation of objects in operational systems nor strategies for retrieval [ETM04].

The MIR systems must now tackle the problems that come from the heterogeneity of the metadata standards and the nature of the datatypes they introduce [TLS04]. Rising from different multimedia communities and embodying different perspectives, the metadata standards define sets of concepts for their domains of activity. However, it is not easy to find mappings between the sets of concepts, even in the cases where overlapping exists.

The nature of the metadata datatypes is another challenge. Vectorial datatypes with hundreds of dimensions such as the Scalable Color Descriptor, a 256-sized vector defined by the MPEG-7 standard, have to be indexed for an efficient use in retrieval.

1.2 Proposal

The goal of this work is to build a multimedia retrieval system on the substantial existing results concerning the analysis of multimedia items and on the concepts and descriptors identified by the standards. Towards achieving this goal, we will present three main contributions. First, a multimedia database model is proposed as a unifying solution for the diversity and complexity of the multimedia items. The model is intended for implementation in a retrieval system supporting preprocessing of the multimedia objects, management by repository managers and search by several kind of users. The MetaMedia model [RD01] accommodates descriptive and content analysis metadata and sets the ground for their integration in search.

The motivation for designing a model is twofold: flexibility and evolution. First, specialists from diverse areas create and need descriptors of different nature. As repositories get more heterogeneous, objects with different creation contexts and different associated metadata must be managed together and we want to keep as much as possible of their original attributes. Second, the effort involved in developing a system for repository management and search is considerable, but new technologies tend to make them obsolete very fast. If the underlying model correctly captures the main concepts, it is possible to incorporate new features and even to redesign the system without starting all over.

The second contribution comes as a consequence of the data model implementation into a real database. More exactly, trying to model the various metadata descriptors for a better search experience, we have encountered the special case of high-dimensional descriptors. Such descriptors are known to be difficult to search with the traditional databases [BBK01, TBM03]. We propose the BitMatrix [CRD06] indexing method and test it using an especially developed evaluation framework called "Multidimensional Multimedia Descriptor Indexing" (MMDI) [GCRD07].

The motivation for a high-dimensional indexing method is the following. As computing power is cheap, it is viable to have image and video processing algorithms analyzing object features and generating large amounts of descriptor data. We want to adopt indexing methods that, besides accommodating the expected diversity of descriptors, can be effective in the basic retrieval operations [Tes04]. Multidimensional indexing requires assumptions on the nature of data and the algorithms to search it. Several requirements from the application domain may condition the choice of the indexing method. A common requirement is that updates to the object database are allowed. This will lead to indexing methods able to incrementally update their data structures. Another important aspect is the ability to add new descriptors. To meet this requirement, it is necessary to allow varying dimensionality in descriptors, and again to be able to extend the indexing structure piecemeal. Handling a large number of multidimensional descriptors may be inconvenient in some parts of the retrieval process. Being able to search on a chosen subset is therefore a desirable feature. The search is usually based on a distance between the query and the candidate items. Useful metrics can take many forms, and the choice of a specific one influences the retrieval process [Agg02, AHK01]. Supporting a wide range of metrics is therefore another important requirement.

The third contribution is a multimedia retrieval system, called MetaMedia which integrates the proposed data model and the BitMatrix index. MetaMedia has been developed as a common platform for loading and exploring multimedia repositories using state-of-the-art technologies for data storage, content rendering and user-interface design.

MetaMedia has been used in the context of two applications with requirements for multi-level structuring, integration of high-level and low-level descriptors, and content-based retrieval. Both applications make use of metadata but they are quite different in nature. The *Enthrone* case is focused on quality of service for news broadcasting and video on demand scenarios in heterogeneous networks. The *Santa Maria da Feira* document center aims at an integrated view of digitized heritage documents, archival descriptions and transcription texts produced by scholars.

Retrieval on video collections has been tested with queries combining text, image and video and the prototype system has been subject to evaluation in the TRECVID [SOK06, SOK04] initiative.

1.3 Overview

The dissertation is organized as follows. The following two chapters offer a presentation of the MIR field. In Chapter 2 we give an overview of multimedia retrieval and present a generic MIR architecture, detailing all the involved components. The chapter starts with a discussion on the current issues in MIR and then surveys multimedia items, queries, databases, retrieval systems and evaluation. The discussion about multimedia items underlines their heterogeneity of presentation formats and justifies the need of a dedicated data model. In Chapter 3 we focus our survey on the essential matters regarding multimedia retrieval: features, similarity measures and indexing methods. The features selection, their quality and the similarity measures are essential aspects of effective multimedia retrieval. At the same time, efficiency is equally important and can only be obtained with indexing techniques that are able to speed-up the overall search process. Chapter 4 introduces MetaMedia's architecture and functionalities. From the architectural point of view, several distinct components are gathered under the MetaMedia name. To illustrate each one's place we instantiate the generic MIR architecture presented in Chapter 2. Then we show that from the point of view of the functionalities, MetaMedia offers browsing, search, administration, content analysis and annotation. Among MetaMedia's components, two are especially important: the data model and the BitMatrix index. The data model is presented in detail in Chapter 5, where we discuss the principles, concepts, classes and the attributes that characterize the model. In Chapter 6 the focus is on high-dimensional indexing in general, and on the BitMatrix index in particular. The chapter has two main parts. In the first one, the BitMatrix index is proposed along with an appropriate similarity criterion. In the second part, we tackle the problem of evaluation for high-dimensional indexing approaches and propose an extensible framework for the observation of such methods, called MMDI (Multidimensional Multimedia Descriptor Indexing). We have observed the time performances, the effect of metric substitutions and the number of distance computations, comparing the BitMatrix index with similar approaches.

While Chapters 4, 5 and 6 are presenting the MetaMedia's architecture detailing the data model and the indexing method components, in Chapter 7 the discussion is focused on the search and retrieval processes. The complex nature of the multimedia items leads to dedicated retrieval approaches that target different parts of them. The MetaMedia's search strategies are integrated in a common user interface accounting for: search on textual content, search on structured contextual metadata and search on image and video features. Chapter 8 presents two case studies that make intensive use of metadata and are both implemented on top of MetaMedia. Very different in nature, the two case studies illustrate the successful use of the data model and of the basic search tasks. Chapter 9 focuses on the retrieval system evaluation. The experiments have measured the quality of MetaMedia as a video retrieval system and took place in the public TRECVID 2007 evaluation [CRD+07]; the final results are reported.

Chapter 2

Multimedia Retrieval

This chapter makes an overall presentation of the multimedia retrieval field. The discussion starts with a presentation of the current state and issues in multimedia retrieval. In the sequel, Section 2.2 presents a generic retrieval system —a three-layered architecture— with components for storage, indexing, similarity assessment, query formulation and visualization. Then, in the next three sections we give details on some retrieval aspects such as multimedia items, queries and multimedia databases. The presence of both content and metadata, strongly connected within multimedia items is important and at the same time challenging for the retrieval systems designers. Metadata plays an increasingly important role in the items life cycle, becoming as important as the content itself. We detail the item-related aspects in Section 2.3. The search queries, detailed in Section 2.4, can be formulated in several ways, including keyword-based, natural language, by example, by sketch, or combinations of them. The answer sets for this range of queries consist of retrievable item parts such as contents, metadata fragments or combinations of them. With such a diversity of retrievable parts and query modalities, ensuring effectiveness and efficiency in retrieval requires proper database technologies. In Section 2.5 we discuss their role in multimedia retrieval. Sections 2.6 and 2.7 review the existing retrieval methods and systems. The final section of this chapter covers multimedia retrieval evaluation aspects.

2.1 Issues in Multimedia Retrieval

Multimedia information retrieval is about searching in information sources available in any modality: text, image, audio, video. Among all modalities, text is a privileged one. With text we are able to precisely express what we mean, either when we produce documents or when we look for them. It is not surprising that research in the IR field has been successful in providing effective methods for searching large volumes of textual data [Sal89, FO95, DDL$^+$90, Sal71].

Beside text documents, we increasingly produce large amounts of audiovisual items such as personal pictures and video recordings, for which we want to ideally have search tools at the level of the text-based ones. But the search in audiovisual datasets is a much more complex problem due to the lack of a direct semantic connection between our information needs and the items.

Some systems adapt the text-based approaches to the audiovisual datasets [SWRS06, HC04]. That is, to index any available text information. For example, in [HC04], speech transcripts [HOdJ07], closed captions, Optical Character Recognition (OCR) and text wrapped around images have been used with satisfactory results. However, such text features do not always capture the semantics of the non-text items. For a web image for example, the text wrapped around it may not be necessarily related to that image.

There are plenty of multimedia items, such as the ones we often create with our electronic devices, that have neither closed captions, nor speech transcripts nor other text information. In such cases, text metadata can only be obtained with annotations, either automatic by means of especially trained concept detectors, or manual [SBD06, YSLH03]. There are problems with both the annotation types, however. While the automatic annotators provide low accuracy, the manual annotation of multimedia content is a subjective and culturally-biased task whose cost makes it not viable for informal or large repositories.

Beside gathering any available text material, multimedia retrieval needs effective methods of using the objects themselves. Such methods, also referred to as Content-based Information Retrieval (CBIR) methods, have the goal of retrieving the objects upon analysis of their actual content features. For image and video objects, automatic analysis of the content features results in low-level descriptors for color, shape, texture or motion. Their use in CBIR systems is discussed in [LSDJ06, SLZ$^+$03, GJ97]; many of the systems reviewed in these works can be found in Section 2.7.

Two main CBIR issues have been identified. First, is the difference between the multimedia items —as digital representations of some real world objects— and the objects themselves. It is important to realize that there is always a degree of fidelity of the representation process which introduces a so-called “sensory gap”. Mistaking the items low-level features for the real object features may have important consequences. For example, a search based on a real object feature values may not find any pictures of it. This issue is even more important in critical application domains such as medicine, where —due to illumination variance for example— the digital representations may be significantly different than the real objects.

The second issue in CBIR is the difficulty of establishing a semantic connection between the low-level features and the concepts that humans provide in their queries. This mismatch is often referred to as the “semantic gap” [GR95, ZG02]. For example, searching with a query such as “find red cars”, by means of color, shape and texture feature values does not uniquely identifies the red cars; several other red objects with car-like shapes such as wagons, trains, toys or buildings are retrieved. Things get more difficult when multimedia items containing “persons walking up stairs” or highly-subjective concepts such as “entertainment” are requested.

2.2 Multimedia Retrieval Architecture

As in conventional Information Retrieval (IR), the MIR tasks are run over a document repository and their purpose is to retrieve all the relevant ones with respect to a given query. To ensure their

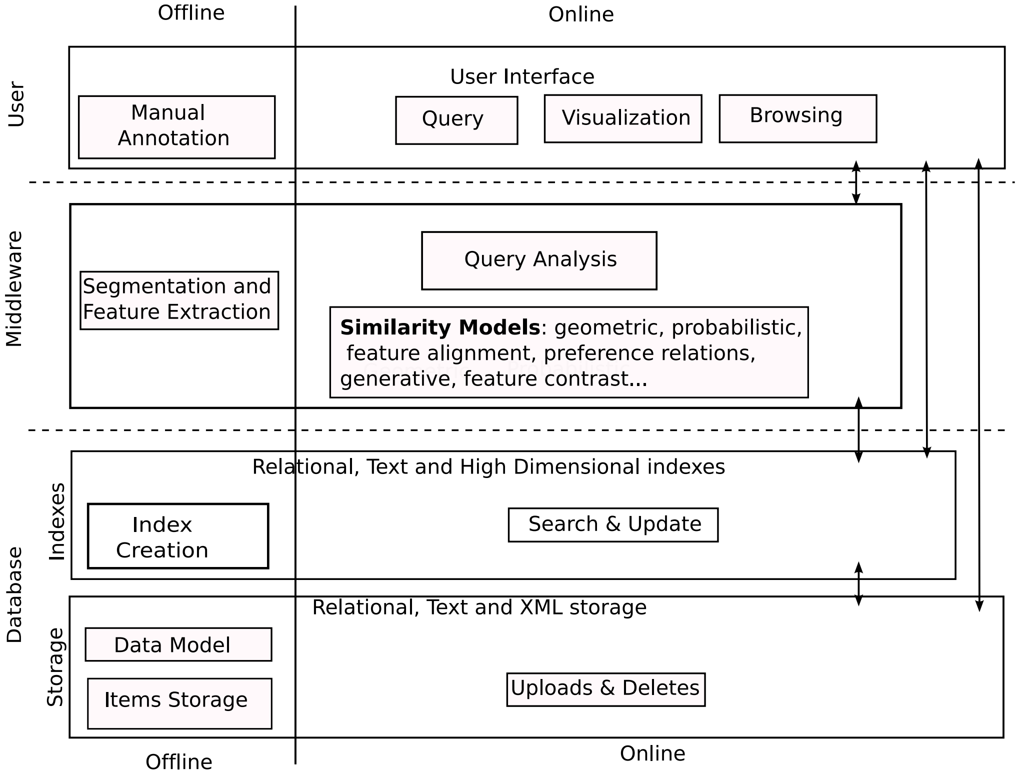

Offline
Online
User
User Interface
Manual Annotation
Query
Visualization
Browsing
Middleware
Query Analysis
Segmentation and Feature Extraction
Similarity Models: geometric, probabilistic, feature alignment, preference relations, generative, feature contrast...
Database
Indexes
Relational, Text and High Dimensional indexes
Index Creation
Search & Update
Storage
Relational, Text and XML storage
Data Model
Items Storage
Uploads & Deletes
Offline
Online

Figure 2.1: Multimedia Retrieval Architecture

functioning, the multimedia retrieval systems glue a wide range of components. The document repositories for instance, require data modeling and storage components. The answer computations are generally performed with similarity assessment and indexing components. From the user perspective, query formulation, query analysis, and results visualization components are also required. Figure 2.1 is an overview of components and tasks in a generic multimedia retrieval system having a three-layer architecture, namely *Database*, *Middleware* and *User*, with typical components and functionalities in each of the layers. There is a loose separation between layers because in concrete retrieval systems, some components may be part of a different layer than the one presented in the figure. For instance, as database systems evolve, more and more middleware functionalities are built into the database layer; for example, some components for similarity assessment may become available directly in the database system.

A different separation, marked by the vertical line in the figure, is between off-line and on-line components. The off-line components are used before the system is up and running, or periodically, while the on-line components are used during retrieval sessions. Typical examples of off-line and on-line components are the ones related to the indexes; their construction and maintenance normally take place off-line, while the search and update are on-line operations.

The *Database* layer is generally responsible for the storage and indexing of the multimedia objects. Its design requires analysis with respect to aspects such as the datatypes, the data model, and the indexing strategies (Section 3.4 reviews the most used indexing methods). For example, it is known that content metadata is of high-dimensional nature and the typical index structures, such as the "R-tree" [BKSS90] do not behave well in such situations; dedicated high-dimensional indexing components could be added to the *Database* layer. Figure 2.1 shows that the *Database* layer has two types of components: storage and indexes. For both there are off-line operations such as the index creation and data model instantiation, and on-line operations such as index search, index update, uploads and deletes.

The *Middleware* layer typically contains components that are neither storage, nor user-related. Nevertheless, middleware components such as similarity models, feature extraction and query analysis are equally important as any other component in a different layer. Among them, the similarity assessment components are especially important because they offer the means for commonly representing the objects and, according to a specific model, to compare them. The most used similarity model, *the geometric model*, requires metric functions [AHK01] over a vector space [BBK01, HAK00], also called *feature space*. However, it has been argued that human similarity judgment cannot always be captured by metrics. Alternative models like *feature contrast model* [Eid03, SJ97] and *preference relations model* [Cho02, BCOO05] have also been proposed; we discuss about them in Section 3.3. Figure 2.1 illustrates the *Middleware* layer with its typical on-line and off-line operations.

At the *User* layer we find components for query formulation, results visualization, relevance feedback [HLJ06, CFB04, IA03, TM03], browsing and manual annotation. The queries, detailed in Section 2.4, can be natural language queries such as "find persons walking up stairs", query-by-keyword queries such as "tennis + court", or query-by-example. The visualization interface is another typical component belonging to the user layer. Minimum functionalities for such an interface include zoom, overview and filter [SBD06]. Another trend in multimedia visualization is faceted retrieval, which allows the user to navigate along conceptual dimensions that are found in the datasets [YSLH03].

2.3 Multimedia Items

Metadata is currently used to assist creation, search, consumption and management activities in a broad range of domains. For some tasks metadata may have a secondary role, but for the others metadata becomes as important as the objects themselves. For example, an image may loose a major part of its relevance for a specific use if is deprived of contextual information such as the authors, terms of use, or technical details. Moreover, metadata itself can be a stand-alone retrievable object [vONH04, NvOH05].

In this section we introduce the *multimedia items* as objects that comprise the raw multimedia contents as well as their metadata. Multimedia items typically have a complex structure. It is common for objects to encapsulate parts in different media, to have references to components they do not directly include, and to have complex relationships among them. This is reflected in the models for multimedia objects adopted in one of the most used metadata standards, MPEG-7 [SKP02].

2.3.1 Metadata

The concept of metadata itself –the data about the data– is hard to define due to the different roles it plays in various communities [KBD^{+}05, Nac04]. The borders between *data* and *metadata* are difficult to establish in environments that include technologies with complex interrelationships. In a library for example, the title, the author and publication date of a book are considered metadata. In the context of a video distribution network these attributes are considered data, while network bandwidth and adaptation parameters are metadata. As another example for the variety of data/metadata definitions, the P/Meta [SS06] standard introduces the "essence" concept for the raw data, the "meta-essence" concept for keyframes and video regions, the "content" concept designates the bundle of essence and contextual metadata, and the "asset" concept designates the content together with its usage rights.

Metadata is used to describe the context in which the multimedia items have been created, are stored or can be used, or to describe aspects of the object content. It addresses several aspects of multimedia items and includes:

- context information such as title, date and authors;
- content analysis regarding details about the content of a multimedia object such as color, texture, shape, or motion activity;
- technical details such as identification issues, size of objects;
- terms of use such as digital rights;
- administrative such as transport, adaptation or custody.

Metadata can be further classified as low-level metadata, such as the color histogram of an image, and high-level or semantic metadata that is concerned with abstractions such as people, animals or buildings.

Another classification of metadata accounts for its source: automatic or manual. There is currently a great interest in automatizing the metadata extraction process [JCLZ05, ZC03].

Finally, metadata is domain dependent. For instance, there may be some specific descriptions concerning the land cover and the relief, that are useful only in geographical applications. Similarly, descriptions such as the number of goals or number of passes are relevant only in sports domain.

2.3.2 Standards

Information communities need to effectively access and share their resources. Metadata standards emerged from digital libraries, audiovisual production and distribution and knowledge representation communities. They address the need to catalog and retrieve on large collections of documents, to interchange and re-purpose content produced in different contexts and to deal with the increasing complexity of the technological platforms used to create, manage and view objects.

In the sequel we will be looking at some well-known standards and identify key aspects from the point of view of a metadata model. This includes considering the application domains, the coverage in terms of descriptive and content-analysis metadata, and the way the concepts relevant for multimedia repositories are captured.

- Dublin Core (DC) [Dub07], a document-centered standard, defines a set of 15 attributes including title, creator, date and subject. DC does not tackle issues related to the structure of documents or collections and considers mostly descriptive information.
- ISAD, ISAAR, EAD [ISA99, ISA04, EAD07] are focused on archival description. They relate documents to the institutions and people involved in their creation and include descriptions for sets of documents, capturing the hierarchical structure of collections.
- MPEG-7 [SKP02], a product of the ISO/IEC, is aimed at providing means for the description of multimedia content that be can of use for search engines, multimedia archives and metadata production tools. It standardizes the *Description Definition Language (DDL)* for the definition of schemes for the description of media and the *descriptors (D)* for different audio-visual low-level features such as color, texture, shape, motion, melody contour. It also introduces a semantic layer of *description schemes (DS)* for structural and conceptual representations of D and DSs.
- TV-Anytime [TV-07] provides a metadata framework for building Electronic Program Guides geared toward broadcast and on-line services. It introduces description elements for programs, parts of programs, groups of programs and real-time information.
- NewsML [New07] is a standard whose aim is to represent and manage the electronic news life cycle: production, delivery and archiving.
- MPEG-21's goal [BdWH+03, MPE02] is to provide a complete framework for the multimedia delivery chain. The standard uses the "Digital Items" as the objects to be packed and distributed. MPEG-21 contains parts such as Digital Item Declaration (DID), Digital Item Identification (DII), Intellectual Property Management and Protection (IPMP), Rights Expression Language (REL), Digital Item Adaptation (DIA) and Digital Item Processing (DIP).
- Material eXchange Format (MXF) is an open file format for the interchange of audio-visual material. It has been designed with the aim of improving interoperability in a content-independent fashion.
- Resource description framework (RDF) and RDF Schema (RDFS) [Con04] represent the W3C's proposals for the general description of information on the Web. RDF is designed to express statements about resources, and constitutes an elementary language for Web metadata interoperability. Concrete descriptors come from vocabularies such as Dublin Core or any of the above. RDFS is a type system for RDF, making the connection to ontology languages.

Table 2.1: Concepts and metadata categories in standards

	Dublin Core	RDF/OWL	ISAD/EAD	MPEG-7	MPEG-21	TV-Anytime	P/META	MXF
General								
Structure	N	Y	Y	Y	core	Y	Y	core
Domain	any	any	archives	AV	AV	TV	TV	TV
Metadata Categories								
Descriptive	core	N	core	marginal	S	core	S	S
Content	N	N	N	core	N	N		N
Technical	Y	N	N	Y	Y	N		Y
Administrative	N	N	Y	N	S	S	Y	N
Main Concepts								
Item	Resource	Resource	Unit of Description	Multimedia Content	Digital Item	Content	Program content	
Segment				Segment	Component		Essence	Essence
Creator	Creator		Creator	Creator	N		Y	
Archive	N	N	Y		Y		Y	
User	N	N	N	Y	Y	Y	Y	
Distributor	N	N	N	Y	Y	N	Y	

- Web Ontology Language (OWL) [AAHH03] is also part of the W3C semantic web architecture concerned with description of information and ontologies. OWL it is also a type system, with similar goals as RDFS.

- P/Meta [SS06], developed by the European Broadcasting Union, addresses the exchanging process of TV programs in industries domains. It defines a business process model with four entities: content creators, content distributors, content repositories(archives) and content consumers. However, only the first three of them are effectively addressed in P/Meta. The fourth, content consumers such as personal video recorders (PVRs), are defined in TV-Anytime which addresses the TV and broadcast industries.

Although comparing standards from different domains is hard, Table 2.1 highlights some of the issues we will be discussing and how they are addressed in the selected standards. The table has separate columns for each of the selected standards and is split horizontally in three parts. The first one treats the general nature of the standard, saying whether it provides constructs for structuring the objects and identifying the domain for which it has been proposed. DC, for instance, does not prescribe structure for the objects while structure is the core of MPEG-21. RDF/OWL are not domain-specific, but MPEG-7 is centered on audiovisual objects and TV-Anytime on television.

The second part of the table analyzes the presence and dominance of some metadata categories. Descriptive metadata, available in most of the standards, is at the core of ISAD, marginal in MPEG-7 and supported (S) by MPEG-21, P/Meta and MXF; these standards do not define descriptive metadata models but assume there will be metadata associated to the objects.

The third part of the table takes the main concepts for generic multimedia repositories and identifies their presence and the adopted terminology in the standards. The basic 'Item' is given different names in the standards. The concept of 'Segment' comes from MPEG-7 but corresponds to the 'Essence' in the TV standards. The concepts of 'User' and 'Distributor' appear in the AV and TV standards and not in the archival standards.

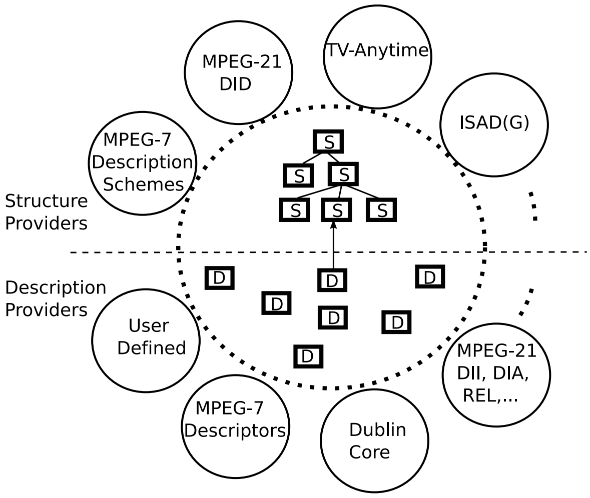


Figure 2.2: Structure and Description Providers

Globally, the table shows that standards from the archival areas provide detailed models for descriptive metadata, and those from the AV and TV areas concentrate on content description and distribution. Standards originated in the television industry are either focused on the file formats, like MXF, or on the business process, such as P/META. Coding aspects have not been considered in the table. They are not central in multimedia databases, where it is assumed that format conversions are needed at import and export time. Currently most standards are supported on XML or provide some XML format as well.

2.3.3 Items

Standards are geared towards the consistent use of a description model and the ease of data interchange. It is important to characterize the elementary objects introduced by standards and to investigate the relations between similar concepts across the main standards.

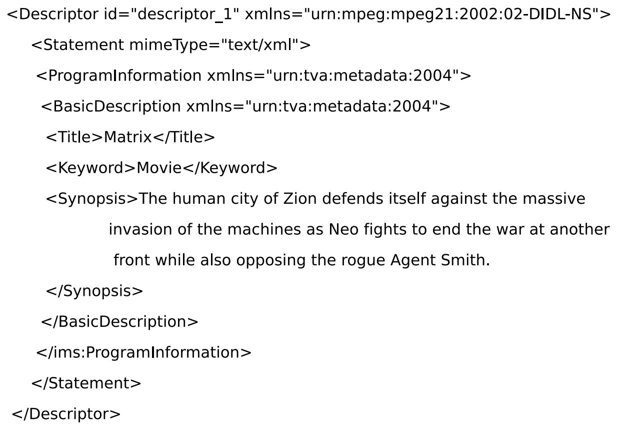
The expression *multimedia item* will be used to capture what is described by different terms in different domains. In the information science area, ‘document’ is the chosen term. ‘item’, ‘object’, ‘content’ and ‘resource’ are used on more technological contexts to denote the subject of description.

An elementary item may be a written text, a photo, a video or audio record. Items may be more complex and aggregate parts that might be different in nature and in media: a bundle of video and sound records, a news item composed of text and photos. In the sequel we will deal with multimedia items that can have a complex structure, possibly aggregating other items; both complex items and their fragments have associated features that can be captured as descriptors.

To account for the structure of complex items and for their atomic description we consider that each multimedia item has a structural part and a description part. Based on this, we classify metadata standards as *structure providers* and *description providers*.

In Figure 2.2 a generic multimedia item is represented as a structure in the spirit of the MPEG-7 Description Scheme, with an associated set of descriptors to capture specific features. Standards such as ISAD, MPEG-21 DID, TV-Anytime, MPEG-7 (DS) and RDF can provide structure, while Dublin Core, MPEG-7 (D), MPEG-21 (DII, DIP, REL, IPMP) can provide the specific descriptors. Within the set of available descriptors, some can be applied to an entire item as well as to segments (parts) thereof, while other descriptors can only be applied to segments.

Given the breadth of the description standards, the same type of information may be available from two or more standards. Dublin Core, MPEG-7, or TV-Anytime can provide a *Title* description for an MPEG-21 Digital Item. The choice of one of these standards as the metadata source is left for the content provider or metadata authoring tool.



```
<Descriptor id="descriptor_1" xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS">
    <Statement mimeType="text/xml">
     <ProgramInformation xmlns="urn:tva:metadata:2004">
      <BasicDescription xmlns="urn:tva:metadata:2004">
       <Title>Matrix</Title>
       <Keyword>Movie</Keyword>
       <Synopsis>The human city of Zion defends itself against the massive
                  invasion of the machines as Neo fights to end the war at another
                   front while also opposing the rogue Agent Smith.
       </Synopsis>
      </BasicDescription>
     </ims:ProgramInformation>
    </Statement>
 </Descriptor>
```

Figure 2.3: Combination of MPEG-21 and TV-Anytime

The multitude of standards creates potential for combinations and this is already being explored. Standards such as MPEG-7, MPEG-21 or TV-Anytime allow the inclusion of metadata from other standards. For example, the MPEG-21 Digital Item declaration Schema, allows `Statement` elements, intended for the insertion of descriptions from other standards. Similarly, the TV-Anytime Phase 1 Schema incorporates metadata from MPEG-7 namespace, while TV-Anytime Phase 2 incorporates MPEG-21 elements as well. Figure 2.3 shows a Descriptor excerpted from an MPEG-21 digital item which includes a TV-Anytime `ProgramInformation` element.

2.4 Queries

The first step toward satisfying an information need is to express it as a query into the search interface of a multimedia retrieval system. Depending on the search modalities available in the search interface, the multimedia queries can be expressed with keywords and natural language topics, but they can also be image examples, video samples, audio files or even combinations of these modalities [CDES05, FSN^{+}95, ISF98, LH04, RTL02, RVM06, RSB^{+}04]. Some systems offer visual interfaces that help the users “sketch” their examples.

Unfortunately, a standard multimedia query language similar, to what Structured Query Language (SQL) is for the data retrieval field, does not exist yet. The complex nature of the information needs, the variety of media modalities and the heterogeneous data models are possible reasons for this situation. In order to illustrate the complexity of query formulations, we take the example of a multimedia query asking for video shots of tennis players on the court, with both players visible at the same time. Such a query can be formulated in several ways. As a keyword query for example, it could look like "tennis +players+court". As a natural language query, its formulation can be "Find shots of tennis players on the court –both players visible at the same time". As a query-by-example, the query can be either a picture that captures two tennis players on the court or a video sequence of a tennis match or an audio file containing racket hits. Finally, a bundle that includes a natural language formulation, a video shot (with or without the audio track) and an image can also be regarded as a query. The last query type —a bundle of data in several modalities— represents the query format that has been adopted in the recent video retrieval benchmarks [SOK06].

Due to the nature of the information retrieval task, there is always a certain gap between the users information need and what the queries really express. It may be caused by subjective factors such as the users ability to express their needs, or the users experience with search systems, or more objective factors such as interface constraints or lack of meaningful examples. In interactive multimedia search systems, the gap can be overcome by helping the user refine the initial queries with the help of relevance feedback techniques [CFB04, HLJ06]. In automatic search systems, i.e. without human interaction during the search, a technique called pseudo-relevance feedback can be applied [Sal71]. That means that the top k documents in the result set are assumed relevant and are used to refine the initial query. Otherwise, if query refinement is not performed, the assumption is that queries are as close as possible to what the users want.

The queries expressed at the interface level have to be analyzed and transformed in order to make them compatible with the internal representation of the multimedia items. In practice, the query analysis process implies almost the same operations that have been done at indexing time, for each of the multimedia items. That means the query analysis processes and therefore their results —the internal query representations — are highly dependent on the retrieval system that they are part of. The best places to look for types of query representations are the presentations of the various retrieval models [Jag06, NNT05, UJ04, ETM04, AAG02, Sal71].

2.5 Multimedia Databases

In this section we talk about multimedia databases and their role in retrieval systems. We start with a set of requirements from the application domains.

2.5.1 Requirements

Retrieving multimedia data efficiently, requires proper storage and indexing strategies. Such concepts are quickly associated with databases mostly because of the out of the box functionalities that they offer: centralized persistence, guaranteed integrity, concurrent access, security constraints and efficient indexes for the common datatypes.

Building reliable multimedia databases is a hard task because of the large variety of modalities (text, video, audio, image) and encoding formats (MP3, MPEG-2/4, MPEG-21) for both the content and the metadata. Often what is consumed, is not only the raw content (an image or a video), but complex items that also contain text and XML metadata. Hybrid databases built to support combinations of relational, XML and generic binary large objects are required. An efficient access to multimedia items also requires indexing structures, generally datatype-oriented, such as numeric indexes, text indexes, XML indexes and high-dimensional indexes. Some datasets require special preprocessing efforts in order to become indexable. Metadata often appears in text/XML format, but the nature of it is rather non-textual, containing numeric feature vectors. For example the MPEG-7 descriptors are embedded in XML but their nature is numerical. In such a case an XML index is useless. The data must be converted to numerical datatypes and indexed with high-dimensional methods [Agg02, GPB04, DN05, CRD06].

2.5.2 Databases

Multimedia databases can be built with different database technologies including relational databases, object databases, object relational databases, native XML databases, hybrid relational and XML databases.

Most of the currently used database systems are based on the relational model [Cod83]. Structuring classical data types based on relations has a sound theoretical background and has proved intuitive and well supported. The data is structured in a set of tables which have rows representing objects and columns representing their attributes. The Structured Query Language, developed for the relational model, permits optimized processing of data by means of insert, update, query and delete operations. The trade-off between execution speed and data redundancy gives the degree of normalization of the relational model. SQL queries on a highly normalized database i.e. having no redundant data, may run slower than on a database that allows some degree of redundancy. The Relational Database Management Systems (RDBMS) provide a natural setup for highly structured data, efficient implementation for the most common data operations, a standard query language, and respond well to the scalability and search flexibility criteria. However, with the ever evolving XML-based formats, frequent changes in the metadata models are expected. The relational model by itself, lacks such a dynamism, extensibility and even portability.

Object-oriented databases [DDB91] store the information as persistent objects which are instances of user-defined classes. This gives the programmer the possibility to encapsulate the data in specific, user-defined ways. Recent data-types such as audio, image, video or other user-defined types can be stored in the database in the same format that the application requires, thus avoiding unnecessary conversions. However, scenarios which require different views and different functionalities on top of the same data might still be difficult to implement.

Taking the best of the relational and object-oriented approaches, the object relational databases [SM95] are suitable for situations where an existing relational model needs to be enriched with object-oriented facilities.

XML databases [WK03] appeared as a natural storage solution to the increasing quantities of data that are being stored and interchanged in XML format [XML07]. There are two categories of XML databases: native XML databases and XML extensions. The Native XML databases [FS04, JAKC+02] use especially designed data types and manipulate data only by means of XML. Although internally they can be built on a relational, or object-oriented database, or a proprietary storage format, no other datatypes are made available: XML documents go in, XML documents go out. On the other hand, the XML extensions enrich the functionalities of the existing database systems by storing XML data either in a text format [Ora04], or through a mapping of XML to relational schema [CCB07, CCB04].

2.5.3 Databases for Multimedia items

Ideally, we look at the databases for multimedia items as typical databases on one side, that is with their standard queries, backup, integrity and security facilities, and on the other side as supporting imprecise IR tasks. A recent research domain, called DB&IR, tackles these issues [Wei07, CRW05]; the results will hopefully contribute to the MIR field.

Since the majority of the multimedia items are represented as XML documents, an XML storage system that supports the current XML Schemas could be preferred. We have already seen the existing XML storage approaches: native XML databases [FS04, JAKC+02] and XML extensions of RDBMS [BCJ+05, CCB07, Ora04]. The native XML databases generally do not support typed representations of the data, other than XML. An array of integers for example, is treated as one string and it is not straightforward to access the individual elements. As multimedia content descriptors, such as the MPEG-7 descriptors presented in Section 3.1, heavily use vector/matrices datatypes, the native XML solutions are not yet prepared for supporting multimedia items.

Another possible approach is to store multimedia XML documents in relational databases. For this, we need to map XML Schemas to database models. The available RDBMS-based XML storage solutions can be classified into two major categories according to their mapping approaches: schema-conscious approach and schema-oblivious approach. In schema-conscious approach, design of the database schema is based on the understanding of XML Schema. [DKD+02] for example, defines a relation for each element and uses primary-keys and foreign-keys to describe the parent-child relationships between elements. In schema-oblivious approach, a fixed database schema is used to store the structure and the data of any documents without the assistance of the XML Schema. There are arguments for and against both the schema-conscious and schema-oblivious approaches. On one hand, it is generally accepted that the schema-conscious approach has better query performance than the schema-oblivious approach since the data is already partitioned based on the XML Schema. On the other hand, the schema-oblivious approach is more efficient for re-constructing data back into XML. The use of custom data types, which is essential to multimedia retrieval, is only supported by the schema-conscious mapping approach. Imposing a fixed model, the schema-oblivious approach does not have such a feature. The work in [CCB07, CCB04] proposes an approach that combines the two mapping types. Based on the observation that the data values are held in leaf nodes, it uses schema-conscious mapping for them. For the internal nodes, schema-oblivious mapping is used.

A hybrid relational and XML database, similar to what has been proposed in [BCJ+05], seems to be the closest to the nature of the multimedia items and can be identified as a possible storage solution. However, the idea of a hybrid relational and XML database by itself is not of much help, unless we define a proper data model for multimedia items and we develop indexing strategies for high-dimensional data.

2.6 Multimedia Retrieval Methods

This section gives an overview of the retrieval strategies that can be found behind the various implementations. Considering the generic retrieval architecture presented in Section 2.2, the real-world multimedia retrieval systems appear as particular cases with various degrees of fidelity to it.

The fact that MIR systems are meant to cope with complex multimedia items and not text-only documents, increases the complexity at each of the architectural levels: user, middleware and database. At user level, beside keyword and natural language-based queries, there can be more query types such as by-example or by-sketch. At middleware and database levels there can be dedicated similarity and storage models, respectively. The number of retrieval strategies could be equal to the number of functional combinations between user, middleware and database components. For example, any of the possible query types could be combined with a similarity and a storage component. However, in practice we have identified three main categories of retrieval methods.

The methods in the first category follow text-based techniques exclusively. They extract from the multimedia objects everything that can be indexed by textual means. Given this approach, the middleware and database layers incorporate text similarity models and text storage, respectively. At the user-interface level the queries are also formulated by textual means: keyword-based or natural language queries. Among the real-world multimedia systems that apply pure text retrieval methods there are “Google”, “Yahoo” and “Microsoft” image search engines and the “mSpace” [SWRS06].

A second category of multimedia retrieval methods are the ones supporting solely query-by-example or by-sketch queries. In these cases, the query expressivity is limited to whatever the examples can offer. For example, in systems that support query-by-sketch, such as [FSN+95, MHH99], the user is requested to transpose its information need into a shape, or a simplistic drawing which is then matched against the items in the database. In systems such as [CTB+99, ORC+97, SC96, MM99, PPS96, CMM+01, EB03b, PBG04, LH04], the query-by-example paradigm is used, but the user must also specify the criterion for the similarity assessment: color, shape, texture or spatial location.

A third category of MIR methods are hybrid. The systems that implement such methods typically start from natural language, or keyword-based queries, and seek relevant multimedia items by using any information available. Quite often, the only available information are the automatically extracted low-level descriptors. In such cases, mapping natural language queries to custom combinations of low-level descriptors becomes the main problem to solve. For example, how can a query such as “Find persons talking on a telephone", be translated into color, texture or shape feature values? Systems such as [RVM06, HW97, IBM08, Moj04] try to overcome such difficulties by applying machine learning techniques with which high-level concepts such as “portrait”, “animal”, “outdoor image” or “building” are automatically extracted. Similarly, [SWG$^+$06] focuses on the automatic detection of concepts, where the concepts are taken from a fixed lexicon [Lar08, NST$^+$06]. However, the main problem with these systems is that their accuracy is quite low; concepts such “entertainment” are especially difficult to annotate. Moreover, the number of possible concepts used by humans largely surpasses the number of available detectors [HYL07].

2.7 Multimedia Systems

In this section a review of the most used multimedia retrieval systems is presented. Surveys on this topic can be found in [DJLW08] and [SW05].

QBIC Developed by IBM [FSN$^+$95], it is one of the earliest image retrieval systems with CBIR query facilities. QBIC supports the retrieval of images based on a number of features including color, texture, and shape. The search interface allows the user to compose a query by drawing the rough shapes and choosing the color of objects according to what the retrieved image should convey. The similarity computation uses a weighted Euclidean distance and the indexing is implemented by using R*-trees [BKSS90]. The resulting images are displayed in a grid sorted by decreasing similarity scores to the query features.

NEC AMORE In this system [MHH97, MHH99] the query process starts by selecting a category of images. An initial set of images can be selected at random or by keyword. Of these images, visually similar images can be retrieved. The images are segmented into at most eight regions of homogeneous color, and downsized to 24×24 pixels. The regions in this picture are directly used for matching either by shape or color. The matching implies a correspondence between regions in the query and target image. The shape similarity between two regions is based on the number of pixels of overlap, while the color similarity between two regions is the distance in HLS space between the colors of the regions.

Blobworld In Blobworld, the images are previously categorized [CTB$^+$99] and the user must start by choosing one of the categories. The features used for querying are the color, texture,

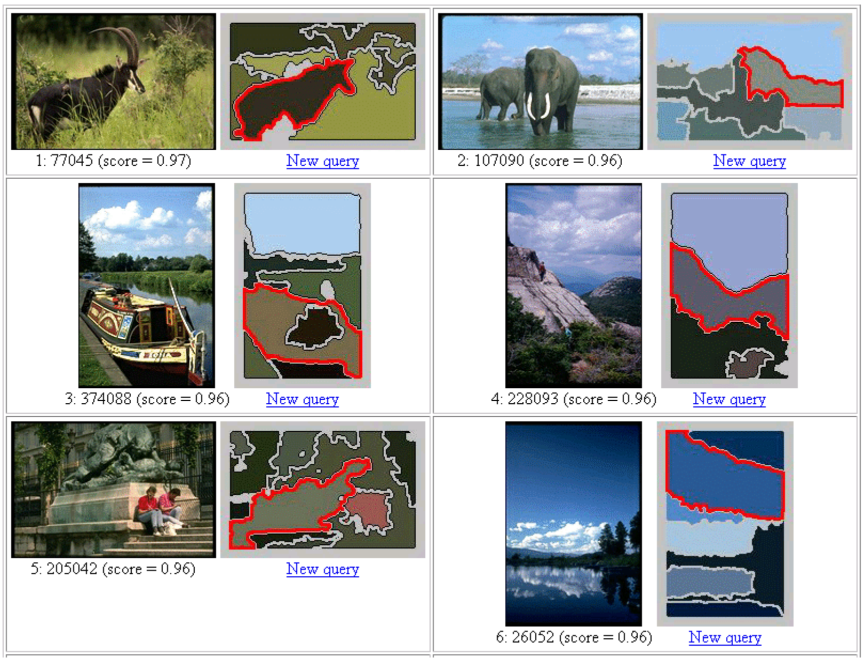


Figure 2.4: Blobworld: queries and results

location, and shape of blobs and of the background. The color is represented by histograms. Texture is represented by mean contrast and anisotropy. Shape is represented by (approximate) area, eccentricity, and orientation. The user selects a blob (an image region), and indicates the importance of the blob with qualifiers such as 'somewhat', 'very'. The user must also indicate the importance of the blob's color, texture, location, and shape. Figure 2.4 is taken from [CTB$^+$99] and illustrates pairs of queries and their best matches.

MARS [ORC$^+$97] supports queries on combinations of low-level features such as color, texture and shape. The desired features can be specified either by example, or by choosing colors or textures from some available pallets. The color feature is represented using 2D histograms over the HS coordinates of the hue, saturation, value (HSV) space. Texture is represented by two histograms, one measuring the coarseness and the other the directionality of the image, and one scalar defining the contrast. The shapes are represented by means of Fourier Descriptors (FD). Interactive query refinement is made possible through relevance feedback.

NETRA [MM99] works on a query-by-example paradigm and targets image regions. For this goal, all the images in the database (the Corel dataset) are segmented into regions for which color, texture, shape and spatial location features are derived. The query images must also come from the database and the criteria must be chosen from color, spatial location, texture, or shape.

Photobook [PPS96] is designed for specific types of content: faces, shapes and textures. It uses separate representations for each of them. To perform a query, the user selects some images from a grid of still images displayed and/or enters an annotation filter.

PicHunter [CMM$^+$01] is designed for image retrieval, based on features such as color histograms and color spatial distributions. The search process starts with a query-by-example paradigm and the similarity is assessed with the L_1 distance (see Section 3.3.3) between individual feature vectors. PicHunter uses a Bayesian framework as the relevance feedback mechanism. It tries to predict the target image the user wants based on the history of the session. The history records the images displayed by the system and user's selections in the previous iterations. A vector retaining each image's probability of being the target is updated at each iteration and the images are displayed in the decreasing order of probability.

VisualSEEK [SfC97] supports text-based and color-based queries through a catalog of images and videos collected from the Web. Color is represented by means of a normalized 166-bin histogram in the HSV color space. The user initiates a query by choosing a subject from the available catalogue or entering a topic. The results of the query may be used for a color query in the whole catalogue or for sorting the result list by decreasing color similarity to the selected item. Also, the user has the possibility of manually modifying an image/video color histogram before reiterating the search.

MIRROR [Po05] is a content-based image retrieval system that offers query-by-example, hierarchical and random browsing; relevance feedback is also supported. It uses several MPEG-7 visual descriptors: Dominant Color, Scalable Color, Color Layout, Color Structure, Edge Histogram and Homogeneous Texture. A new similarity measure, the Merged Color Palette approach, is developed for the Dominant Color descriptor.

IBM-Marvel [IBM08] applies machine learning techniques to detect high-level concepts in video data from automatically extracted audiovisual features. It automatically associates confidence scores to the assigned concepts and organizes semantic concepts using ontologies. Keyword and natural language queries are supported.

Google, Yahoo, Microsoft These image retrieval systems support keyword-based queries and the answer computation is based primarily on the text metadata.

Vizir [EB03b] is an MPEG-7 based visual information retrieval framework which focuses on similarity measurement with application-specific features (color, texture, shape and motion) and query acceleration based on R-tree indexes [Eid03, Eid02].

ImageGrouper is a CBIR system with an interaction strategy [NMH02]. The emphasis lies on group-based search, as this system combines the tasks of searching, annotating, and organizing images by groups. By dragging images around a workspace, i.e. in and out of groups, and selecting different groups as negative or positive examples, the user gives relevance feedback information without having to think in terms of the system's internal representation.

CalPhotos In [Cal] there are a number of alphanumerical attributes available for querying, such as the collection, keywords, location, county, photographer. The colors of each image are quantized into 13 colors bins. Six values are associated with each color bin: the percentage of the image with colors in that bin, and the number of "very small", "small", "medium", "large", and "very large" dots of that color found. Image features are stored as text strings. For example, a picture of a sky with clouds might have a few large white regions, and a large amount of blue, and would have a feature text string "mostly blue, large white few". Search is performed by sub-string matching.

Informedia The Informedia system [HW97, WKSS96] offers content-based multimedia retrieval based on the analysis of the entire spectrum of modalities: native text, texts results of speech recognition, image, audio and video. The search process is a collaborative interaction of image, speech and natural-language queries, which has the advantage of compensating erroneous or ambiguous formulations that may appear if a single query modality would be used.

2.8 Multimedia Retrieval Evaluation

The merits and usefulness of a retrieval system can only be revealed by proper evaluation methodologies. A first objective of the evaluation efforts could be to measure the user satisfaction, but this would require an expensive and exhaustive test scenario. For instance, in such an experimental setup, a large set of users would have their behaviors monitored. The multitude of variables such as execution speed, user's search skills and the learning effects are a serious impediment for such an evaluation scenario. However, recent efforts such as the Video Olympics [SWdR+08] might represent a step toward measuring user satisfaction. An alternative to the multitude of variables is to create evaluation systems at smaller scales, where the possible sources of variability are easily controlled.

Another evaluation direction could be towards measuring the execution speed, i.e the efficiency. However, no such tests exist because the efficiency was considered marginal in the IR world. But with the growth of multimedia mass production, scalability problems do appear, and we expect that retrieval evaluation will focus on efficiency issues also.

The current evaluation efforts are towards measuring the quality of retrieval, i.e the effectiveness. In [TS92] and then in [Wes04], the authors distinguish between system-oriented tests and task-oriented tests. System-oriented tests measure if a system works properly given a specific similarity criterion and task-oriented tests evaluate the usefulness of a system functionality for a given task. Considering for example, the case of similarity with respect to the "DominantColor" descriptor for the image retrieval task. A system-oriented test will evaluate if the retrieved images are really similar based on color information, while a task-oriented evaluation will try to find how useful the images is for a specific information need. Both the system-oriented and task-oriented evaluations take place in the same environment which is defined by 4 components: a fixed number of multimedia objects to be searched on (dataset), a fixed number of topics from which the queries will be chosen, evaluation criteria such as the precision/recall measure, and the relevance judgments or ground truth. Each of these components need to be tuned for a successful evaluation and have been subjects of research. Regarding the collection's content nature, it makes sense to use datasets with relevant provenance. For example, retrieval systems dedicated to medical images should be tested on datasets of medical provenance, but no one imposes that. It is not known what is the optimum size of the collection of multimedia objects. There are also no specifications on what the number of topics should be. Regarding the evaluation measures, they depend on the existence of relevance judgments (ground-truth). Such judgments may not come from an objective authority (and usually don't), therefore they are influenced by subjective assumptions.

We observe that there are still many variables, although fewer that in an user satisfaction evaluation. Therefore, it is natural to question the reliability of the current evaluation tests [HL05]. Especially developed significance tests show that the current evaluations are still dependent on datasets, queries and ground truths [HL05, SZ05]. Most of the multimedia retrieval systems presented in Section 2.7, claimed successful in their times, were tested on datasets such as COREL, BRODATZ, BENCHATHLON or COLUMBIA which cover specific domains. Such datasets are clearly clustered into subsets such as *buildings, buses, beach, mountains, red flowers, white flowers*, and it is relatively easy to retrieve objects belonging to one of the subsets. Flowers, for example, are difficult to be mistaken for mountains, even if the retrieval processes uses basic low-level features. Little can be said about the performance of such systems in broader and more heterogeneous datasets.

The current MIR evaluation efforts, such as the TREC/TRECVID [SOK06, Wes05, OLIG04] and CLEF/IMAGECLEF, consist of task-oriented tests that measure retrieval performances from large and heterogeneous collections. TRECVID for instance, consists of a fixed set of video shots, a fixed set of topics (queries), a fixed set of relevance judgments and performance measures. Different retrieval approaches are submitted by various participants, and have their relative performances measured. Each of the approaches produces a ranked list of documents for each topic. The quality of each ranked list is measured per-topic, based on the positions of the relevant documents in it. For this, the Average precision (AP) is used, where the precision is measured at every rank at which a relevant document is obtained and then averaged over all relevant documents to obtain the average precision:

$$AP = \frac{\sum_{r=1}^{N} P(r) \times relevant(r)}{size\,of\,ground\,truth}.$$

r is the rank, N is the number of retrieved documents, $relevant()$ is a binary function on the relevance at a given rank, and $P()$ is the precision at that rank. If a score for the whole retrieval system is needed, then mean average precision (MAP) is generally used, where MAP is the mean of the AP scores for all topics.

Chapter 3

Multimedia Analysis

In this chapter we continue the state of the art in the MIR field by giving more insight on three aspects, namely feature extraction, similarity models and multidimensional indexing. We consider these three aspects essential for a successful retrieval. The feature extraction and the similarity models are important for the accuracy (effectiveness) of the retrieval, while the multidimensional indexing account for efficiency. It is important to note however, that although we present them jointly, these aspects are not always tightly coupled. Each one contributes to several distinct application domains, among which multimedia retrieval is just one of them. For example, multidimensional indexing, as a database-related aspect, is required in data retrieval, on-line analytical processing (OLAP), and recently in multimedia retrieval. Similarly, feature extraction is required in recognition, diagnostics or classification tasks, and examples exist for the similarity models as well.

The first section of this chapter covers the feature extraction aspect as the process of obtaining properties from multimedia items. We show that there can be directly obtained properties, also called low-level features [Bob01, DKN04, HR05a, MM96, MFKMT$^+$00], such as color or texture, and high-level features obtained with automatic analysis of the low-level feature data [JNY07, JCLZ05]. The feature extraction results are typically incorporated as descriptors, which according to the feature types, can be low and high-level. The low-level descriptors, consisting predominantly of vectorial data, have little or no meaning for the human perception. The high-level ones are human concepts from any knowledge domain.

In the second section we show that the similarity models are used with the purpose of discriminating between multimedia items. As the items cannot be directly used, the similarity is computed based on some common representations [ML, EB03a, FKS03, FTCTF01, LLL01].

The final section reviews the multidimensional indexing methods which are required when high-dimensional datasets have to be searched. In the case of multimedia datasets, with descriptors easily surpassing hundreds of dimensions, the high-dimensional indexes have been increasingly required. Various approaches [BBK01, CNBYM01, HS03] have been proposed, but they have limiting data pre-processing requirements and they work with some fixed metrics only. The most versatile, but time consuming technique, is the sequential scan. Searching efficiently on these datasets is still a difficult task.

3.1 Features and Descriptors

Feature extraction is the process of obtaining properties from multimedia items. When the items are digitized versions of some real world objects, it is important to note that the items properties are not necessarily the properties of the objects in the world. Section 2.1 has briefly explained that the digitizations are obtained by sensory means and the sensors limitations introduce a so-called "sensory gap". For example, due to lighting conditions, a red car can appear as white in an image item. It is easy to imagine, that a color-based search for multimedia items that contain red cars would not find that digital item. Although this is a trivial example, in plenty of critical domains such as medicine, the "sensory gap" must be seriously taken in consideration. However, throughout this section we are not concerned with this issue; no relationships between real world objects and multimedia items are assumed.

In a first, and most referred categorization, the features can be either *low* or *high-level*. The *low-level features* are the ones directly obtained from the multimedia items, such as color, texture, shape, motion and audio features [DKN04]. They are used in several application domains namely object recognition, surveillance, diagnostics and content-based retrieval. The *high-level features* consist of human concepts from any knowledge domains. They are either manually obtained with human annotation efforts, or with automatic annotators that are trained on the low-level feature data [JCLZ05, ZC03].

Beside the low/high-level categorization, the features can also be *local/global* and *variant/invariant*. If the feature extraction targets specific item regions, the features are called *local* and if the whole item is analyzed, the features are called *global*. A feature is considered either *variant* or *invariant* if the feature values are sensitive or not to item transformations. For example, if the shape feature values of an image are insensitive to rotation, such a feature is called rotation-invariant. The variance/invariance can be judged with respect to any other distortions, such as scaling, location change, illumination variation, viewpoint transformations, or occlusions.

The feature representations are typically incorporated as descriptors. According to the feature types presented above, the descriptors can be considered low/high-level, local/global or variant/invariant. Between features and descriptors there is a one to many relation. Thus, a feature can be represented in multiple ways. For example, DominantColor, ColorStructure and ScalableColor descriptors [MOVY01] are all color descriptors. The low-level descriptors consist predominantly of vectorial data. The DominantColor descriptor for example, consists of an RGB-tuple (red, green, blue). The color histogram descriptors are even larger vectors of 128 or 256 values. The entire set of descriptors induces a vectorial space, also called the feature space, which generally has hundreds or thousands of dimensions.

A possible question to address in this section is: what descriptors should be extracted that would help finding relevant multimedia items [JNY07, DKN04]? As an answer to such a question, one may expect a set of descriptors that guarantees up to some degree of confidence that a proper retrieval system can be built on top of it. Such an answer is difficult to obtain, because the retrieval quality is a problem that surpasses the choice of descriptors. However, we know that retrieval based on low-level features is already mature [FSN$^+$95, SC96, RSB$^+$04, WKSS96], but inferring high-level features is still a challenging task [Han06, XZTT06, GS04, HOdJ07, SWS^{+}00]. As the high-level feature extraction depends, up to some extent, on the low-level features, we expect that the use of low-level features will only grow.

3.2 High- and low-level descriptors

As representations of the high-level features, the high-level descriptors consist of keywords or more complex natural language formulations that capture human concepts. For the text-based multimedia items, high-level descriptors can be directly extracted from the contents themselves, but for the other modalities a direct semantic connection to human concepts does not exist. In the second situation, high-level descriptors can be obtained by means of either automatic [JCLZ05, ZC03] or manual annotations. Whatever the annotation type, there are arguments for and against their use. For instance, the manual annotations are considered subjective and expensive to obtain, but if domain experts validate the annotations, their accuracy is valuable [SBD06, YSLH03]. An argument in favor of automatic annotations is that are cheaper to obtain when trained concept detectors already exist. However, such detectors are available only for a small number of concepts and often provide low-accuracy rates. If concepts such as “water”, “sky”, “cars”, “faces” or “outdoors” are relatively well-detected, concepts such as “entertainment”, preferences [BCOO05] or moods [Han06] are far from being correctly identified. To cope with the automatic concept detection challenges, the multimedia communities are currently establishing concept lexicons [NST^{+}06, SWvG^{+}06], focusing on the concepts that are feasible for automatic detection [HYL07, SWG^{+}06, YH06]. Even though the detectors have low accuracy rates, the use of large concept sets of approximately 5000 concepts has proved beneficial for video retrieval [HYL07].

The remaining part of the current section presents low-level descriptors. Most of them are included in the MPEG-7 standard. MPEG-7 can describe still pictures, graphics, 3D models, audio, speech, video and their combination. It is also possible to describe content that is not audio, nor visual such as the taste or smell [SKP02]. Beside the MPEG-7 descriptors, there are other types of descriptors such as color anglogram, gabor texture, tamura texture or SIFT (scale-invariant feature transform) descriptors.

3.2.1 Color Descriptors

DominantColor The “DominantColor” descriptor specifies up to eight colors of an image and their spatial coherency parameter which is useful for similarity matching. Dominant colors can be specified for a whole image or an arbitrarily local region. The next sample shows a “Dominant-Color” descriptor:

```
<VisualDescriptor xsi:type="DominantColorType">
 <ColorSpacetype="RGB"/>
   <ColorQuantization>
      <Component>R</Component>
      <NumOfBins>8</NumOfBins>
      <Component>G</Component>
      <NumOfBins>8</NumOfBins>
      <Component>B</Component>
      <NumOfBins>8</NumOfBins>
   </ColorQuantization>
   <SpatialCoherency>31</SpatialCoherency>
   <Value>
      <Percentage>15</Percentage>
      <Index>255 255 255</Index>
      <ColorVariance>1 1 1</ColorVariance>
   </Value>
</VisualDescriptor>
```

ColorLayout This descriptor specifies the spatial distribution of colors. To calculate the color layout of an image, the image is separated into 64 equal parts using a 8x8 raster. For each segment the most representative color is determined. The Euclidean difference of each color is used for image matching. The next sample shows a “ColorLayout” descriptor:

```
<VisualDescriptor xsi:type="ColorLayoutType">
   <YDCCoeff>27</YDCCoeff>
   <CbDCCoeff>20</CbDCCoeff>
   <CrDCCoeff>49</CrDCCoeff>
   <YACCoeff5>12 9 13 21 13</YACCoeff5>
   <CbACCoeff2>10 12</CbACCoeff2>
   <CrACCoeff2>24 16</CrACCoeff2>
</VisualDescriptor>
```

The “ColorLayout” descriptor uses the YCbCr color space with quantization to 8 bits using a Discrete Cosine Transform (DCT) algorithm. In the YCbCr color space format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value. A more detailed description of the “ColorLayout” descriptor and how its parameters are calculated can be found in [SKP02].

ScalableColor The “ScalableColor” descriptor specifies a color histogram in the HSV color space, which is encoded by a Haar transform. Typical applications of the “ScalableColor” are image search. The descriptor also forms the basis of the *group of frames/ group of pictures* descriptor [MFKMT+00]. A sample is presented below:

```
<VisualDescriptor xsi:type="ScalableColorType"
                  numOfBitplanesDiscarded="0" numOfCoeff="64">
   <Coeff>-202 71 27 54 -7 -1 7 14 6 13 11 22 -2 3 10 14 0 1 0
          2 -1 5 0 0 -6 -2 1 5 -15 5 1 -4 0 0 0 1 0 0 1 2 1 1
          1 3 1 2 4 5 1 -3 2 -2 -3 -3 -9 -7 0 -15 -15 -15 -14
          -15 -18 -15
   </Coeff>
</VisualDescriptor>
```

Color Structure The “Color Structure Descriptor” is a histogram which aims at identifying localized color distributions using a small structuring window. It counts the number of times a specific color is contained within the structuring window, as this visits the entire image. The size of the structuring window scales with the image size. For images smaller than 256x256 pixels, an 8x8 raster is used and than grows according to a predefined rule [MOVY01].

```
<VisualDescriptor xsi:type = "ColorStructureType" colorQuant = "4">
 <Values>
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  65  117  119  112  104  104  106  109  116  120
  122  121  124  124  124  122  117  113  113  115  114  105  89  73
  45  8  2  2  0  0  0  0
 </Values>
</VisualDescriptor>
```

3.2.2 Texture Descriptors

Texture features are intended to capture the granularity and repetitive patterns of surfaces within a picture. For instance, grass, land, brick walls, and flower petals have each different texture patterns. The texture features are important in domain-specific image retrieval, such as aerial and medical imaging. For the latter, they are especially close to the underlying semantics. MPEG-7 includes at this time, three texture descriptors, namely the “Texture Browsing”, the “Homogeneous Texture” and “Edge Histogram”. Beside these we also present the Gabor and Tamura texture descriptors [MOVY01].

Texture Browsing This is a compact descriptor that requires only 12 bits (maximum) to characterize a texture’s regularity (2 bits), directionality (2 bits), and coarseness (2 bits). A texture may have more than one dominant direction and associated scale. For this reason, the specification allows a maximum of two different directions and coarseness values. The regularity of a texture is graded on a scale of 0 to 3, with 0 indicating an irregular or random texture. A value of 3 indicates a periodic pattern with well-defined directionality and coarseness values. The directionality is quantized to six values ranging from 0° to 150° in steps of 30°. Coarseness is related to image scale or resolution. It is quantized to four levels, with 0 indicating a fine grain texture and 3 indicating a coarse texture [MOVY01].

Homogeneous Texture The "Homogeneous Texture" descriptor [KIR+01] applies to images and is obtained from both the images themselves and their Fourier transforms. First, the mean and standard deviation values of the original images are obtained. Then, the frequency space (the Fourier transform of the images) is partitioned into 30 channels, for which energy and energy deviation values are calculated. The final form of the descriptor is:

$$[I_{mean}, I_{sd}, e_1, e_2, .., e_{30}, ed_1, ed_2, .., ed_{30}],$$

where the I_{mean} and I_{sd} represent the mean and standard deviation values of the original image pixels and the e_i, $ed_i, i \in \{1, .., 30\}$ are the energy and energy deviation values obtained from the i^{th} frequency channel. A sample is presented below:

```
<Descriptor xsi:type = "HomogeneousTextureType">
   <Average>56</Average>
   <StandardDeviation>196</StandardDeviation>
   <Energy>
    240  228  227  239  198  206  216  187  196  212
    202  193  190  173  154  206  161  173  162  136
    127  176  144  136  161  122   97  144  112  124
  </Energy>
  <EnergyDeviation>
    242  228  224  238  197  205  203  177  192  207
    186  183  178  166  135  195  143  166  157  110
    121  163  138  132  157  104   81  141   88  113
 </EnergyDeviation>
</Descriptor>
```

Edge Histogram The "Edge Histogram" descriptor captures the spatial distribution of edges. The computation of this descriptor is fairly straightforward. A given image is first sub-divided into sub-images, and local edge histograms for each of these sub-images is computed. Edges are broadly grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic (non-orientation specific). Thus, each histogram has five bins corresponding to the above five categories. The image partitioned into 16 sub-images results in 80 bins. These bins are non-uniformly quantized using 3 bits/bin, resulting in a descriptor of size 240 bits. A sample is presented below:

```
  <Descriptor xsi:type = "EdgeHistogramType">
    <BinCounts>
     0  1  1  4  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
     4  3  2  6  1  0  3  1  3  2  0  4  0  0  1  0  4  2  0  1
     3  4  2  3  2  4  3  3  4  4  3  3  2  4  3  3  6  4  3  3
     0  6  4  4  3  1  6  4  4  3  1  5  5  5  4  0  6  5  5  3
   </BinCounts>
  </Descriptor>
```

The “Edge Histogram” is especially useful for image matching when the underlying texture is not homogeneous [MOVY01].

Gabor texture descriptor It is obtained from an original image by applying the Gabor wavelet transform at several scales and orientations [MM96]. Two values are calculated at each transform: the mean and standard deviation values of each transform coefficients. For example, for an input image, four possible scales and six possible orientations, the descriptor will be a vector of length $4 \times 6 \times 2$:

$$[\mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}, .., \mu_{46}, \sigma_{46}].$$

More details about the Gabor texture descriptor can be found in [MM96].

Tamura texture descriptor Tamura [TMY78] devises texture features that, according to the authors, correspond to human visual perception. Six textural values namely the coarseness, contrast, directionality, line-likeness, regularity, and roughness are calculated [HR05a].

3.2.3 Shape Descriptors

Region Shape The region-based shape descriptor expresses pixel distribution within a 2-D object region; it can describe complex objects consisting of multiple disconnected regions as well as simple objects with or without holes. It uses a complex 2-D Angular Radial Transformation (ART) [RCB05], defined on a unit disk in polar coordinates. The default region-based shape uses 35 coefficients quantized to 4 bits/coefficient [Bob01]. A sample is presented below:

```
<Descriptor xsi:type = "RegionShapeType">
 <MagnitudeOfART>
   15  15  11  13  13  15  7  15  10  11  8  8  2  7  12  7
   9  13  4  10  7  6  8  8  1  8  7  2  5  10  5  8  9  4  3
 </MagnitudeOfART>
</Descriptor>
```

3.2.4 Motion Descriptors

Camera Motion This descriptor characterizes 3-D camera motion parameters. It is based on 3-D camera motion parameter information, which can be automatically extracted or generated by capture devices. The camera motion descriptor consists of the following camera parameters: fixed, pann, track, tilt, boom, zoom, dolly, and roll [JD01].

Motion Activity The “Motion Activity” descriptor captures the intuitive notion of intensity of action or pace of action in a video segment. The intensity of activity is encoded as an integer lying in the 1-5 range. A value of 5 indicates a high level of activity and the value of 1 indicates the lowest level of activity. This descriptor [JD01] is useful for applications such as video repurposing, surveillance, fast browsing, dynamic video summarization, content-based querying etc.

3.2.5 Audio descriptors

The list of most used sound descriptors includes: Spectral Centroid, Spectral Roll-of Point, Spectral Flux, Compactness, Spectral Variability, Root Mean Square, Fraction of Low Energy Windows, Zero Crossings, Strongest Beat, Beat Sum, Strength of Strongest Beat, MFCC (Mel Frequency Cepstral Coefficient), LPC (Linear Predictive Coding) and Area Method of Moments. Some examples are show below. More can be found in Appendix A.

```
<feature>
  <name>Spectral Centroid Overall Standard Deviation</name>
  <v>8.81E-1</v>
</feature>
<feature>
  <name>Spectral Flux Overall Standard Deviation</name>
  <v>1.363E-4</v>
</feature>
<feature>
  <name>Root Mean Square Overall Standard Deviation</name>
  <v>4.869E-3</v>
</feature>
<feature>
  <name>MFCC Overall Standard Deviation</name>
  <v>1.538E0</v>
  <v>2.536E-1</v>
  <v>2.873E-1</v>
  <v>4.181E-1</v>
  <v>2.302E-1</v>
  <v>2.241E-1</v>
  <v>2.108E-1</v>
  <v>2.2E-1</v>
  <v>1.853E-1</v>
  <v>1.798E-1</v>
  <v>1.889E-1</v>
  <v>3.092E-1</v>
  <v>1.908E-1</v>
</feature>
```

3.2.6 Miscellaneous descriptors

Anglograms The main idea is to extract image regions and represent them as *feature points*. The regions can be identified either by means of shape or color. The *anglograms* are obtained in two phases. First, the Delaunay triangulation for the set of feature points is computed and then, from the angles of the Delaunay triangles an angle histogram is computed [ZG02].

Keypoint features It is a particular type of feature which identifies image keypoints which are expected to be invariant to geometric changes such as scale and rotation and photometric changes such as addition of noise and change in illumination. Details about the keypoints selection can be found in [Low04]. The keypoint features have been successfully used to obtain descriptors such as the bag of features (BoF) [JNY07]. The idea behind BoF is to gather all the keypoints across the dataset and cluster them. The cluster centroids are treated as “visual words” in a “visual vocabulary”. By mapping the images to the “visual word-space”, vector representations are obtained for them.

3.2.7 Discussion

There are several parallel descriptor categorizations, namely low/high-level, local/global, and variant/invariant. There can be descriptors that are in the same time low-level, local and invariant, or high-level and global, or any other combination. Recent years have been marked by a shift from global and variant descriptors, such as color histograms and global shape descriptors, to local invariant ones, such as keypoints, region-based, and local shape characterizations. This shift is explained by the complexity of the multimedia items. The semantics covered by a whole item is too deep for global descriptors to cope with. It has been observed that local descriptors correspond better to item parts such as objects or persons. Invariance also became increasingly important because it helps identifying objects under various circumstances.

3.3 Similarity Models

The similarity models are the means by which the multimedia items can be compared. Although the choice of a specific similarity model is generally dependent on the available feature types, in this section we look at the similarity models independently of this type of dependency. The main question has a general nature: how can one discriminate among multimedia items? In order to find out, we start with a preliminary discussion on what similarity is, and then continue with a review of several similarity models.

3.3.1 About similarity in MIR

The similarity between two multimedia items can be seen generically as a relationship between them. There are domains, such as geometry, in which the similarity is precisely defined: two geometrical objects are called similar if one is congruent to the result of a uniform scaling (enlarging or shrinking) of the other. With this definition we can easily asses the similarity between geometric objects. For example, two circles are always similar to each other, two squares are always similar to each other, and two triangles are similar if and only if they have the same three angles. But unlike geometry, in the MIR field it is not clear yet how to precisely define the similarity between multimedia items.

In [KZB04], the authors stress the difference between similarity and matching applications, stating that they imply different approaches. Matching is a simple comparison between two items, —a binary operator— that checks whether the items are identical or not; it is used for copy identification, for example. But similarity, although it sometimes uses matching techniques, is a more complex process, requiring knowledge from very diverse domains like biology and psychology. In their efforts for modeling the similarity, the authors of [SJ97, Gen88] observe that the similarity judgments are many times based on subjective features grouping which yield user-dependent degrees of similarity.

Currently, in MIR systems the similarity is evaluated by some sort of comparison between the items feature values (descriptors). The features that can be checked for relatedness range from low-level ones such as color or shape, to high-level and subjective ones such as a feelings or moods [Dim04, Han06]. Suppose, for instance, that the "Color Layout" descriptor (see Subsection 3.2.1) has been extracted and the items are described by vectors with the following form: [YDCCoeff, CbDCCoeff, CrDCCoeff, YACCoeff5, CbACCoeff2, CrACCoeff2]. Based on this assumption, the similarity can be regarded as the distance between the points in a six-dimensional space. Note that this assumption requires the use of a distance metric (see Appendix B). It is not clear however, whether the similarity relationship should be of metric nature or not [SJ97]. There can be other similarity assessment methods, such as the feature contrast model [EB03a], that capture in differently the relationships between multimedia items.

The similarity assessment methods are typically called similarity models. Although they are sometimes called "text similarity models" or "image similarity models", the similarity models are not restricted to specific modalities. The adoption of a specific model depends on the capacity of representing the multimedia items in that model. However, simply being able to represent images into a text similarity model for example, does not automatically guarantees good retrieval results. Whether the model is effective or not, can only be determined with evaluation tests. In the sequel we present the most used models for similarity.

3.3.2 Text retrieval models

The first two similarity models that we present, namely the vector space and probabilistic models, are often referred to as "text retrieval models" because they have been widely used with text datasets [YH07]. Assuming a text retrieval context, the goal for each of these models is to find text documents that are relevant for a given, usually keyword-based query.

The vector space models In Vector Space Models (VSM) [FO95, VR79, Sal71] both the queries and the documents are represented in high-dimensional vector spaces, often called *term spaces*. In such vector spaces each dimension represents a term, where the terms are features that characterize the documents at content level; often (but not necessarily) the terms are the equivalent to the words in the documents. For example, in a term space with N dimensions, document D_k would be represented by vector $D_k = [d_{k0}, d_{k1}, .., d_{kN}]$ and query Q vectors $Q = [q_0, q_1, .., q_N]$, where the d_{ki} and q_i are weights for the i^{th} term. The collection of all the document vectors can be arranged into what is often called a document-by-term matrix which contains rows corresponding to documents and columns corresponding to the terms. The simplest representations are binary vectors, $d_{ki}, q_i \in \{0,1\}$, representing whether the terms are present or not in the corresponding document or query. More complex vector representations can be obtained by assigning positive real weights that encode term frequency information [Sal71]. One of the most popular weighting schemes is called *tf.idf* [Sal89], and proposes that term weights should be proportional to the frequency of the term occurrence within a document, and inversely proportional to the number of documents where the terms appear. The similarity between query and documents can be computed using the cosine similarity: $sim(Q, D_k) = \frac{Q \bullet D_k}{|Q||D_k|}$.

Latent Semantic Indexing Inspired from the VSM, the Latent Semantic Indexing (LSI) [MKS] starts also with a term space representation of the documents, followed by a dimension reduction technique, called Singular Value Decomposition (SVD). As the term spaces are high-dimensional, the intuition is that the data contains dependencies and can be approximately represented in a space with fewer dimensions. Co-occurrence of terms in various text documents is assumed as an indication of dependence, or synonymity. LSI projects co-occurring words onto the same dimension, and independent terms onto different dimensions. Thus, the synonyms get clustered (actually projected) in the same dimension. By selecting a subset of dimensions of the projected space, LSI obtains an approximation of the original term space; the more dimensions, the better the approximation. The query is also treated as a document, thus it has a corresponding vector in the reduced dimensional space. The similarity between a query and any of the documents $sim(Q, D_k)$ is computed also with the cosine similarity, where Q and D are the query, respectively the document vectors in the reduced space. The result set, i.e. the relevant documents, are obtained by sorting the vectors in decreasing order of cosine similarity values.

The VSM-based models in general, and LSI in particular, were widely used in text retrieval systems such as [Fol90, LB97], and with further enhancements in [Dum90, Dum91]. However, there are also problems with the VSM models. The main issue is that there is no real theoretic basis for the assumptions of the term spaces and weighting schemes. Moreover, once a term space is chosen, its term dimensions are not really orthogonal. That happens because the terms are not independent of all the other terms.

Probabilistic Models The Probabilistic Models (PM) try to overcome the problems observed in the VSM such as the reliability of the term space and the choice of term weighting schemes. The idea of PM is to predict the probability that a given document will be relevant to a given query. They rely on the assumption that the distribution of terms throughout the collection, or within some subset of it, may tell something about the likely relevance of any given document. With this assumption, accurate estimates of the probabilities can be obtained and the documents can be ranked according to this probability of relevance. This technique is called the *Probability Ranking Principle (PRP)* [Rob97]. More precisely, the PRP suggests sorting documents by the log-odds of relevance, where the relevance is estimated given the distribution of terms in relevant and irrelevant documents. For a given document D and a query Q the log-odds of relevance R can be expressed like

$$logodds(P(R=1|D,Q)) = log\frac{P(R=1|D,Q)}{P(R=-1|D,Q)},$$

where $P(R=1|D,Q)$ is the probability that D is relevant to Q and $P(R=-1|D,Q)$ is the probability that D is irrelevant to Q. Probabilistic models have been used in [MO06, LMO+96, H.R91].

Language-based models A special category of probabilistic models, the Language-based Models (LM) [Kra05, ZL01, PC98] attempt to statistically model the use of language in a collection in order to estimate the probability that a query is generated from a particular document. The main idea is that, if the query could have come from the document, then that document is likely to be relevant. The LM approach practically consists in associating each document with a language model which is a probability distribution over all the terms in the collection. Formally, for a query Q, given the language model of document D, the conditional probability is defined as:

$$P(Q|D) = P(q_1,q_2,..,q_n|D) = \prod_{j=1}^{N} P(q_j|D_j).$$

The document ranking is then obtained by computing the conditional probabilities of Q given the language model of each document.

3.3.3 Geometric model

Like in VSM, the multimedia items are represented by their coordinates in a vector space and the the similarity between them is computed with metrics, i.e. distance measures, that satisfy the four metric axioms: (1) self-similarity, (2) minimality, (3) symmetry and (4) triangle inequality. The detailed metric axioms can be found Appendix B. The vector space, here called feature space, is induced by the feature types. Assuming a number of F features, each feature contributing with f_i coordinates (dimensions), the total dimensionality of the feature space is $N = \sum_{i=1}^{F} f_i$; an item x in such a space is represented as $[x_1,x_2,..,x_N]$.

A large number of distance metrics have been proposed in the literature on multimedia retrieval. That is explained by the fact that the choice of similarity measure has proved critical to retrieval performance [AKJ02], which justifies the efforts towards finding "better" metrics. Given two items x=$[x_1,x_2,..,x_N]$ and y=$[y_1,y_2,..,y_N]$, several ways for computing the distance $d(x,y)$ are presented in the sequel.

Quadratic Distance Is given by

$$d_A^2(x,y) = (x-y)A(x-y)^T = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(x_i-y_i)a_{ij}(x_j-y_j)}, \qquad (3.1)$$

where A is a similarity matrix, with components a_{ij} representing cross-similarity weights for dimensions i and j in the underlying vector space. By specifying appropriate weights in the A matrix, the similarity distance functions are adapted to specific application requirements or user preferences. Adaptive similarity with the help of quadratic distance has been used in [BKS01a, ISF98, CTB+99].

Minkowski Distance of order p Frequently called the L_p norm, this distance metric is given by:

$$d(x,y) = \{\sum_{i=1}^{N}(x_i - y_i)^p\}^{\frac{1}{p}}. \tag{3.2}$$

Its special cases, namely L_1 (the absolute distance) and L_2 (Euclidian distance), have been widely used in multimedia retrieval. L_1, given by $d(x,y) = \sqrt{\sum_{i=1}^{N}|x_i - y_i|}$, has been considered one of the most effective metrics in practice [YH07]. However, when used with high-dimensional visual descriptors, L_1 and L_2 are outperformed by the L_p norm with fractional values for p ($p \in (0,1)$) [HR05c, AHK01].

Chebyshev Distance

$$d(x,y) = max_i|x_i - y_i| \tag{3.3}$$

Canberra Distance

$$d(x,y) = \sum_{i=1}^{N}(\frac{|x_i - y_i|}{x_i + y_i}) \tag{3.4}$$

Cosine Correlation

$$d(x,y) = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N}(x_i)^2}\sqrt{\sum_{i=1}^{N}(y_i)^2}} \tag{3.5}$$

Cosine Correlation is widely used in text retrieval models such as the VSM and LSI. An example of an LSI-based image retrieval method can be found in [ZG02].

Earth Mover's Distance It has been initially proposed for transportation problems. For example, given the distribution of a mass of earth and a collection of holes, the EMD measures the least amount of work needed to fill the holes with earth. The adaptation of this metric to image retrieval [RTG00] treats similarity as a signature matching problem. A similarity match between two items x,y is calculated by transferring the values of the N dimensions of x to those of y. (x would be the distribution of earth mass and y the distribution of holes). Similarly to the quadratic distance, the EMD metric requires a cost matrix $C = [c_{i,j}]$ to describe distances between dimensions. The matrix elements $c_{i,j}$ reflect the distances between dimensions i and j, also called ground distances. Formally, the computation of the EMD between x and y finds a flow $F = [f_{ij}]$ that minimizes the overall cost of transportation:

$$Transportation(x,y,C) = min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} f_{ij}. \tag{3.6}$$

Once the optimal flow is found, the EMD is defined as the transportation cost normalized by the total flow:

$$EMD(x,y,C) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} f_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij}}. \tag{3.7}$$

This metric was used for image retrieval based on color and texture [RTG00, WB04]. Its benefits were observed also in [DJLW08].

Localized Similarity Functions The goal of this type of functions is to examine only a small percentage of the data. The similarity between two items x and y ($Sim(x,y)$), is obtained by considering only the locality of x in each dimension, where the locality is defined dimension-wise, either as the set of k nearest neighbors, or points in the same range with x_i. The similarity measure formalized in [HR05b] is defined only for the set of dimensions i that contain y_i in the locality of x_i:

$$Sim(x,y) = \sum_{i \in K(x,y,k)} (1 - \frac{d(x_i,y_i)}{z}), \tag{3.8}$$

where the set $K(x,y,k)$ represents the set of dimensions i where y_i lies in the same locality with x_i; K is called a *proximity set* [AY00]. The function d is a local distance function that is applied dimension-wise (only for the dimensions in K), while z is a normalizing factor. The metric defined above is in fact a generalization of the metrics that have been used in [AY00, Cha03] and showed improved ability in using only a small fraction of the dataset.

3.3.4 Feature Contrast Model

The Feature Contrast Model (FCM), also known as Tversky's model [Tve77], is a similarity model that appeared from observations that questioned the metric nature of the human perception and especially the triangle inequality and symmetry axioms.

Unlike the geometric similarity model, which represents the items as points in a vector space, Tversky treats them as sets of binary predicates. Let X and Y be two feature sets corresponding to items x and y. The contrast model obtains the similarity of the two items by combining the common features ($X \cap Y$) and the distinctive features ($X \setminus Y$ and $Y \setminus X$):

$$Sim(x,y) = f(X \cap Y) - \alpha f(X \setminus Y) - \beta f(Y \setminus X). \tag{3.9}$$

It can be observed that the formula in equation 3.9 is not a metric, because the symmetry axiom does not hold. Tversky's view on similarity assessment was extended in [SJ97] and [SJ99] to geometric, respective fuzzy feature contrast models. Comparisons with the Euclidean distance [EB03a], and with various other metrics for MPEG-7 descriptors [Eid03] have shown that in many cases the FCM performs better. However, in [RMBB89] the FCM has been criticized for not capturing relationships between features.

3.3.5 Aggregation model

In this model the assumption is that several independent rankings with respect to a query object already exist and they have to be merged into a single results list. An example of a situation often encountered in practice, is when several high-dimensional descriptors are involved in the similarity computation and metrics adapted to each descriptor are required; we end up with separate rank lists for each descriptor.

Generic aggregation algorithms The common approach for the aggregation of several rank lists is to use scoring functions, such as *min* or *average* in order to compute overall scores. There are two phases:

1. the first phase is common to all of the algorithms in this category, and produces several ranked lists.

2. the second phase combines these ranked lists using various score aggregations; a termination condition is checked.

Depending on the aggregation function and the termination condition, several aggregation algorithms such as *Fagin's Algorithm*, *Threshold Algorithm*, *medrank* [FKS03] and *Quick-Combine* [GBK00], have been proposed.

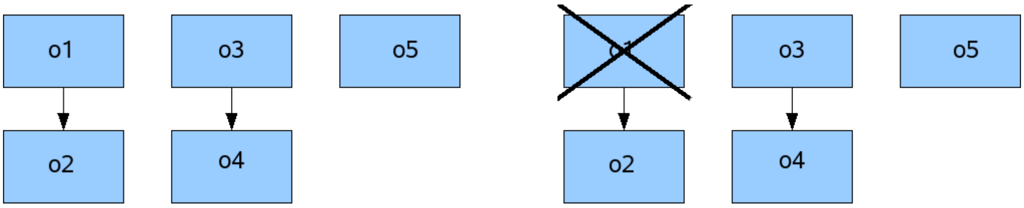
Kendall-optimal and Footrule-optimal aggregations Considering λ the list of all item identifiers and N a number of independently produced rank lists $\lambda_i, i \in \{1,..,N\}$, the idea behind the Kendall-optimal aggregation [FA] is to find the optimal $\lambda_{optimal}$ that could have generated the $\lambda_1,..\lambda_N$ lists. The items order in $\lambda_{optimal}$ is considered the final similarity order, i.e the item placed first would be the most similar, the item placed second would be the next similar, and so on.

The problem of finding $\lambda_{optimal}$, as the closest list to $\lambda_1,..\lambda_N$, is put in the following way: $\lambda_{optimal}$ is the list that minimizes the

$$\sum_{i=1}^{N} d(\lambda_{optimal}, \lambda_i), \tag{3.10}$$

where d is a distance metric between any two lists. Given that $\lambda_{optimal}$ and $\lambda_i, i \in \{1,..,N\}$, are all permutations of λ, a distance metric between permutations, such as the *Kendall tau* metric (detailed in Appendix B), can be used to estimate $d(\lambda_{optimal}, \lambda_i)$. In such a case, the computation of $\lambda_{optimal}$ is referred to as the *Kendall-optimal aggregation* [DKNS01].

If the *Kendall tau* distance is replaced with the *Spearman footrule* distance [DB82, FKS03], the computation of $\lambda_{optimal}$ is referred to as the *Footrule-optimal aggregation*.



(a) Before o_1 removal

(b) After removing o_1

Figure 3.1: Preference Diagram

Aggregating fuzzy information In some cases the items feature values are assigned in a fuzzy fashion. For example, the degree of redness can vary between 0 (not at all red) and 1 (totally red). Similarly, the items roundness attribute can follow a fuzzy degree assignment. Under the assumption that sorted lists, also called *graded sets*, for attributes such as redness and roundness are obtained independently, their aggregation is obtained with fuzzy aggregation functions [Fag99], which are rules that combine them and assign overall scores [Fag02, RNS02].

3.3.6 Preference Relations Model

The scoring functions used in aggregation models are quantitative approaches that assign scores to multimedia items based on their feature values. When multiple rankings exists, the user preferences for one or another result set is modeled with weights. However, the scores and weights have a limited expressive power, since not all the user preferences can be translated into quantitative expressions. The preference-based similarity model appeared as an alternative to this limitation.

Figure 3.1(a) illustrates a first example. A small database composed of five objects is assumed, where object o_1 is preferred to object o_2 and o_3 is preferred to o_4. There is no preference between o_1, o_3 and o_5. If we try to capture these preferences with a scoring function, the object scores can be assigned in the following manner:

$$S(o_1) = S(o_3) = S(o_5) > S(o_2) = S(o_4). \tag{3.11}$$

Let's assume now that object o_1 is deleted from our database (Figure 3.1(b)). Looking at the preferences we see that in the absence of o_1, o_2 should be retrieved, but looking at the scores we have $S(o_3) = S(o_5) > S(o_2) = S(o_4)$, which do not place o_2 among the top objects. That happens because the scoring function evaluates the similarity only by quantitative means, not accounting for the relations with the other objects in the database.

As an alternative to scoring functions, in [Cho03, Cho02] the authors propose a similarity approach based on qualitative preference relations. This technique requires that, given two objects o_i and o_j, there must exist a binary relation —the preference relation— that states whether o_i is preferred to o_j ($o_i \succ o_j$) or not ($o_i \not\succ o_j$). If $o_i \succ o_j$ we also say that o_i *dominates* o_j. If neither $o_i \not\succ o_j$ nor $o_j \not\succ o_i$, then we have that $o_i \sim o_j$, which represents the *indifference relation*. Assuming that queries are expressed by means of preference relations, it is possible to define an operator, called *skyline-operator* [BKS01b] or *winnow* [Cho02], that computes the answer by picking the most preferred objects, i.e. all the objects that are not dominated by the others. The skyline operator is defined as:

$$skyline(DB) = \{o \in DB | \nexists p \in DB, p \succ o\}. \tag{3.12}$$

In the previous example, after the deletion of o_1, o_2 becomes a skyline object, i.e. is not dominated by any other object. A second example, illustrated in table 3.1, consists of set of books described with three attributes, namely *Id*, *Description* and *Price*. Let us assume that between any of these books —represented as tuples $(Id, Description, Price)$— the following preference relation is defined:

Id	*Description*	*Price*
772691	Being and Time	20
772691	Being and Time	15
772691	Being and Time	18
820591	The metamorphosis	5
374164	Descartes' Error	25

Table 3.1: A collection of books

$$(Id, Description, Price) \succ (Id_1, Description_1, Price_1) \equiv Id = Id_1 \wedge Price < Price_1 \tag{3.13}$$

This preference relation can be also regarded as a query and states that if we find books having the same identifier, then we prefer the ones that have a smaller price. However, it does not state anything about the books having different identifiers. According to the preference we have exemplified (Equation 3.13), books such as "The metamorphosis" and "Descartes' Error" find themselves in an *indifference* relation with the any of the "Being and Time" copies. If the *skyline-operator* (Equation 3.12) is used to answer this preference-based query, the set of "best books" shown in Table 3.2 is obtained. From the three copies of "Being and Time", the one with the smallest price is picked. Further, if we iterate the skyline operator on the remaining books, a set of "second-best books", shown in Table 3.3, is obtained.

Id	*Description*	*Price*
772691	Being and Time	15
820591	The metamorphosis	5
374164	Descartes' Error	25

Table 3.2: The skyline books

Although we have illustrated only two iterations, in preference-based MIR applications such as [BCOO05, LK02], several skyline iterations can be performed and the partial object sets can be ranked: the set of skyline objects, followed by the skyline of the remaining objects, and so on.

Id	*Description*	*Price*
772691	Being and Time	18

Table 3.3: The set of second-best books

Although it is a ranking without scores, also called a qualitative ranking, the interest in having retrieval systems that take into account the user preferences led to an increasing adoption of skyline-based similarity models [GSG05, YLL$^+$05, PJET05, PTFS05, BC05, BG04, KRR02, TEO01].

3.3.7 Generative Model

In this model, every object is considered as the outcome of a process that generated it. The idea is to include in the similarity assessment knowledge about the likelihood of existence of the objects to be compared. In [KBT05] it is argued that the generative processes are important since they help the selection of critical features for similarity comparison. As an example, in [KBT05] the authors ask which is more similar to a given nutritious mushroom: a mushroom identical except for its size, or a mushroom identical except for its color? They suggest that knowing how mushrooms are formed, i.e. their generative process, we can be sure that mushrooms grow from small to large and their final size depends on the amount of sunlight and soil fertility. Therefore, it is more likely that the differently-sized mushroom is more similar than the differently-colored.

When applied to multimedia retrieval, the generative models follow probabilistic approaches (see Section 3.3.2). Under the assumption that each item was obtained from some specific generative process, the similarity is assessed through the probability that the query is an outcome of the item's process [Wes04].

The range of applicability for the generative models is not very large for the time being, but they seem to be promising for high-level similarity judgments. For instance we can consider two images, each containing a human face. Based on common features such as shape, color or salient points they may appear very different, but analyzing their "history" or the processes that produced those faces we may find out that they capture the same person.

3.3.8 Network Model

The idea behind this model is to represent the items as nodes in a network, with *part-Of* or *isA* relations between items. The most representative class for network models are the semantic networks, where the nodes of the network are concepts, eventually coming from predefined ontologies. Such a representation mode is often referred as a conceptual graph [NC06]. The measure of similarity between two nodes(concepts), is the length of the shortest path between them. Usually the semantic neighborhood of radius r of a concept C is defined as the set of all the concepts that have the distance to C smaller than r.

The work in [ETM04] proposes a semantic model dedicated to video retrieval with entities and relations between them. Instantiations of this model are graphs, allowing for graph-based queries, while the similarity is obtained by graph-based matching. In [RMBB89] the authors propose a metric on semantic nets and in [RE03] similarity between concepts from different ontologies is investigated. As ontology-based similarity has been increasingly used, there are already database systems that provide embedded support for it [DCES04, CDES05].

3.3.9 Structure-mapping Model

Similarity between objects is some times revealed through analogies. People make analogies that help them show similarity in some respect, like the *processor-brain* analogy for example. Features like color, size, shape or material could hardly bring some relevance for the processor-brain analogy, but the fact that processors and brain both coordinate and maintain the main activities represents the essence of this analogy. Another example from [Gen88] considers the simple arithmetic analogy $3:6$ and $2:4$. The authors state that is not the overall number of features that 3 has in common with 2, nor 6 with 4 that is relevant for the analogy, but the *relation* "twice as great as" that exists between 3 and 6 and also between 2 and 4.

In [Gen88] the authors suggest that the similarity, seen as analogy, is a process of structural alignment and mapping. In their similarity model the items are treated as sets of atomic entities. Further, the model captures the structural representation of the objects by observing various relations between the atoms. The similarity between two objects is obtained by aligning the atoms and the relations across the two objects. The structure mapping engine in [FFG89] is used as a similarity model.

3.3.10 Hybrid Models

Many similarity assessment approaches combine techniques from several models in an effort to improve the MIR quality. In [Sch05] the authors introduce a hybrid model for semantic similarity that combines the network model, in the form of semantic network, with the geometric model. The resulting model becomes a network of vector spaces, where each node of the net represents a concept, and each concept is further mapped to a vector space. The similarity is obtained in two phases: first the query concepts are aligned with the concepts available in the model using semantic networks techniques and then a metric is applied on the corresponding vector spaces. A similar approach can be found in [Rau04].

3.4 Multidimensional Indexing Methods

We have seen in the previous section that the similarity between multimedia items can be evaluated with a wide range of models. In this section we will assume a geometric model, where the multimedia items are modeled as points in a vector space and the similarity between them is assessed with metric-based search operations. Given a query object q from a universe of objects $\mathscr{O}$ and a metric function d, particular sets of objects are of special interest:

- the *Nearest Neighbor Set* $NN(q)$, defined as $\{o \in \mathscr{O} | \forall v \in \mathscr{O}, d(q,o) \leq d(q,v)\}$;

- the *k- Nearest Neighbors Set* $NN_k(q)$, defined as the set of k elements closest to q in $\mathscr{O}$, i.e. a set objects $A \subseteq \mathscr{O}$, such that $\mid A \mid = k$ and $\forall o \in A, v \in \mathscr{O} - A, d(q,o) < d(q,v)$;

- the *approximate Nearest Neighbor Set* ($NN_A(q)$), which is the set of objects $\{o \in \mathscr{O} | d(q,o) \leq (1+\varepsilon) * d(q, NN(q)), \varepsilon > 0\}$.

The computation of these neighbors sets becomes challenging when dimensionality increases, a problem often referred to as the "curse of dimensionality" [BGRS99, IM98]. In practice, simply comparing the query vector to all feature vectors —an approach considered näive and inefficient— has proved comparable, sometimes even more efficient than especially designed indexing methods such as the "R-tree" [BKSS90]. Due to the increasing importance of search in high-dimensional spaces, a great diversity of indexing methods have been proposed in recent years. However, in spite of their diversity, a common philosophy can be observed. That is, to avoid looking at every object by creating groups of objects with common properties. Under the assumptions that the grouped objects are requested or pruned together, checking the groups instead of objects saves time. For example, the objects in a distance range can be pruned or not just by checking the range's minimum and maximum distance limits. In the following we review the most used multidimensional indexing methods.

3.4.1 Spatial Access Methods

Spatial Access Methods (SAM), also called feature-based methods, partition the space based on the values of the vectors along each independent dimension. When a feature space does not exist, one can be constructed in order to make SAM applicable. For example, when the only available information is the distance between the objects, also called distance-only data, a feature space can be built by deriving "features" purely based on the inter-object distances. Such methods are called *embedding methods* [FL95, FO95]. On the other hand, when the feature space is high-dimensional —which is often the case in practice— dimensionality reduction techniques can be applied [Jol86, SZZ07].

Spatial Access Methods (SAM) are based on tree data structures with two types of nodes: data nodes (the leaves) and directory nodes. The information stored in directory nodes describe space regions obtained with various partitioning strategies. There can be strategies such as data partitioning (DP) which uses minimum bounding regions (MBR) such as *R-tree*, *R^*-tree* [BKSS90], *X-tree* [BKK96], bounding spheres such as $SS - tree$ [WJ96], MBR and bounding spheres such as $SR - Tree$, generic minimum bounding regions (hyper rectangle, cube, sphere) such as the $TV - tree$, the $BBD - tree$ [AMN+98] and space partitioning methods (SP) such as the $kDB - tree$, $Hybrid - tree$, $SH - tree$ [BBK01].

We illustrate here the *R-tree*, a representative for data partitioning approaches, that has proved efficient for low-dimensional spaces (2, 3, 4 dimensions). The left-hand side of Figure 3.2 was obtained by running a Java implementation of R-tree found at [Had]; the MBRs around 10 data points $s_1, .., s_{10}$ can be observed. The tree structure on the right-hand side of Figure 3.2 follows the containment hierarchy obtained from the data partitioning.

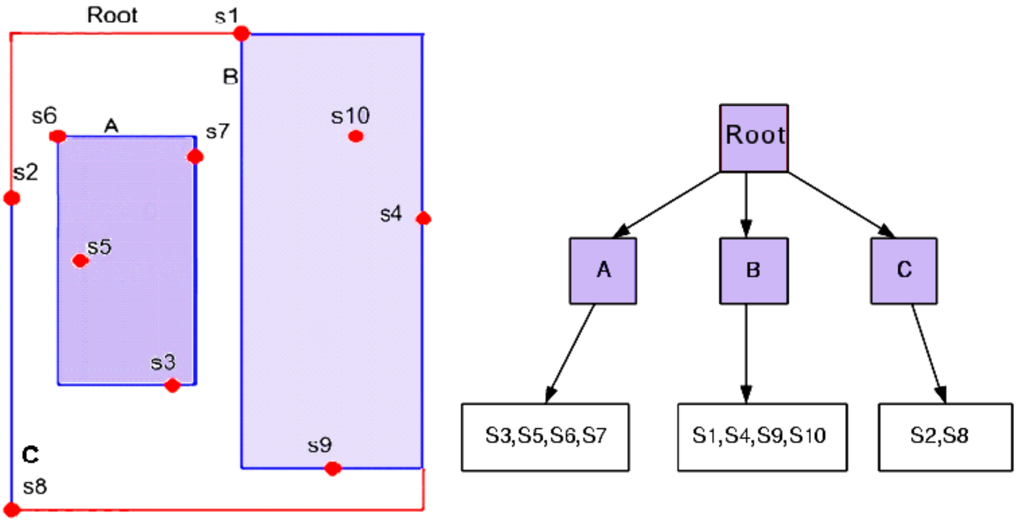


Figure 3.2: An R-Tree example

The nodes of the SAM trees contain information, such as coordinates, about the MBRs that they cover. This kind of information grows exponentially when increasing the dimensions, leading to the growth of each tree node and of the index itself. The bigger the nodes are, the fewer can be stored in a disk page. Note that the disk pages have fixed amounts of space set by the operating system. The number of nodes per disk page is called the *fan-out* of the tree. As the fan-out decreases with the dimensionality, accessing such an index becomes more difficult. Another important issue is the *high-overlapping* between the MBRs stored at the same level in the tree. Although they cannot be observed in our example because we have few data points and few dimensions, the overlappings lead to an increased number of branches to be searched. The costs of maintaining SAM structures cover aspects such as space, index re-creation, updates, insertions and MBRs split managements.

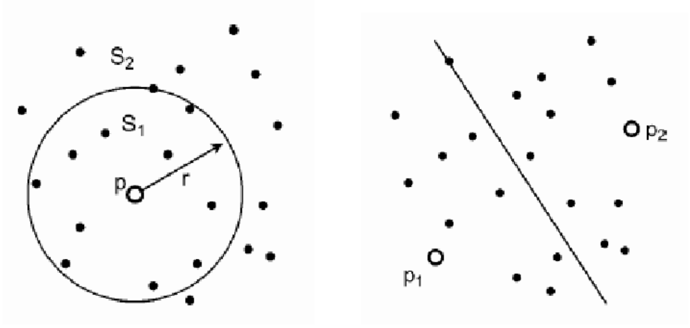


Figure 3.3: Ball and Hyperplane partitioning; figure taken from [HS03]

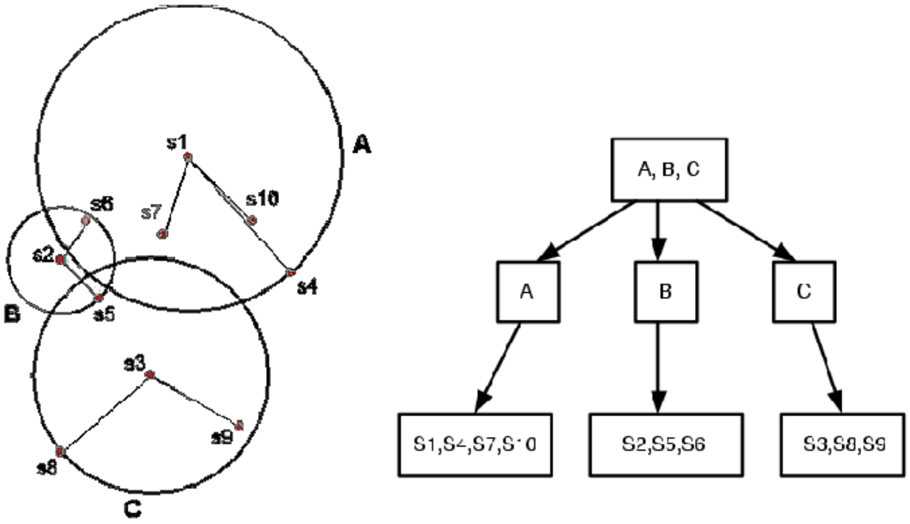


Figure 3.4: An M-Tree example

3.4.2 Metric Access Methods

Like, SAMs, the Metric Access Methods (MAM) are also based on tree-structures, but they work with relative distances between the objects rather than their absolute positions in space. MAM [HS03, CNBYM01] have gained an important role due to the fact that conventional spatial approaches stop being efficient in high-dimensional data. They are also required for search in distance-only data sets, i.e. that cannot be mapped to vectors spaces. An example thereof, is a set of text documents that use the *edit metric* (see Appendix B) to measure the distance between documents.

MAM build their tree-structures by recursively partitioning the data set into subsets at each node level [CNBYM01]. Two main partitioning schemes have been denoted by Hjaltson and Samet [HS03]: *ball partitioning* and *generalized hyperplane partitioning*. Figure 3.3, taken from [HS03], illustrates them. With the *ball partitioning* approach, the data set is partitioned based on the distance from one specified object, called *vantage point* or *pivot*; two subsets are generated: the first subset inside the ball around the pivot, and the second subset outside the ball. Among ball partitioning trees the most referenced are *Vantage-Point Tree (VP-tree)*, *Multi-Vantage Point Tree (MVPT)* [BO99], *Vantage-Point Forest (VPF)*, *Burkhard-Keller Tree (BKT)*, *Fixed Queries Tree (FQT)*. With the *hyperplane partitioning* approach, at each step two points p_1 and p_2 are selected. The elements closer to p_1 than to p_2 go into the left sub tree and those closer to p_2 go into the right sub tree. Among hyperplane partitioning structures we enumerate *Bisector Tree (BST)*, *Generalize Hyperplane Tree (GHT)*, *Geometric Near-neighbor Access Tree (GNAT)*. and the *M-tree*[CPZ97].

An M-tree example for a small set of 10 objects is shown in Figure 3.4. The objects are stored in the leaf nodes, while the internal nodes, also called *routing objects*, store pointers to child nodes and the covering radius for the children they "enclose". The applicability of the MAM methods ranges from "native" distance-only datasets to high-dimensional datasets for which conventional SAM are no longer efficient [DN05]. Their advantage is that the relative distances between objects can be pre-computed at index creation time, avoiding heavy distance computations at search time. However, this advantage is also a constraint because the distance measure used at creation time must be also used at search time. Given that the distance measure must be established in advance, searching with user-defined metrics that capture personalized criteria becomes difficult. It has been shown, however, that metrics from certain classes of parameterizable distance functions can be used [CP02].

3.4.3 Single-Dimension Mapping

Single-dimension mapping approaches map the points in the high-dimensional space to single-dimensional values for which efficient techniques such as the B-tree [BM72] exist.

Querying high-dimensional data in single-dimensional space is proposed in [YBOT04] and [OTYB00], by considering the smallest and largest values among all the dimensions of each data point. Another single dimensional mapping method sorts the data points according to their positions on a space filling curve (Hilbert or z-curve) [LLL01]. The list obtained in this way is stored in a B-tree structure. In practice, several shifted copies of the data points (maximum $N+1$, where N is the space dimensionality) are used. Each copy of the data points produces a separate, differently ordered list, which is stored in a separate B-trees. The left part of Figure 3.5 illustrates a space filling curve that touches the original data points. On the right part, the same curve touches the shifted data with one unit up and one to the right.

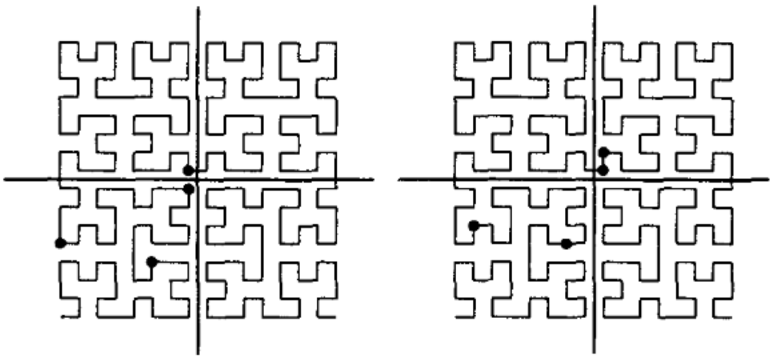


Figure 3.5: Space filling curve; figure taken from [LLL01]

In "iDistance" [JOT^{+}05] a data partitioning (see Section 3.4.1) technique is initially applied, followed by a single-dimensional mapping within each region. The mapping process consists of sorting the objects in each region on the distance to a specific reference point, such as the region's center. The reference points are then indexed in a $B^{+}-tree$ structure. In [SZZ07], the "iDistance" is applied on local subspaces previously obtained with principal components analysis [Jol86].

3.4.4 Aggregation Methods

These methods treat each dimension as a separate list, and their goal is to obtain the answer of the query by accessing a minimum number of lists and as few objects as possible in each of the visited lists.

Branch and Bound on Vertically Decomposed Data (BOND) [AMNK02], adopts vertical decomposition as storage organization. That means it decomposes the data into multiple tables, one for each dimension. Therefore, the information for a specific object is distributed into multiple tables. The algorithm accumulates the distances between the query object and all data vectors, by scanning the dimensional projections one-by-one. After processing a few dimensions, *partial* distances of k-nearest neighbors are exploited to discard safely from further consideration those vectors that cannot possibly participate in the result. This process is graphically illustrated in Figure 3.6. It can be observed that groups of 8 dimensions are successively scanned. After reading each group, based on partial distances, a set of objects is pruned. The iterative application of this process quickly reduces the candidate set (the upper part of the figure) to just a small database sample.

3.4.5 Data Approximation Structures

The methods described here are based on a result obtained in [WSB98], which states that the MAM and SAM are outperformed by a simple sequential scan whenever the dimensionality is above 10. Therefore, their initial assumption for high-dimensional data is that a sequential scan is inevitable. Under this assumption, the VA-file method [WSB98, TM03] constructs a vector of approximations, significantly smaller than the original data and sequentially scans it. The vector of approximations is the result of a space division in a number of cells; all the objects contained in a cell are represented by a common approximation. When searching for nearest neighbor for example, the entire vector of approximations is scanned. Based on their minimum/maximum distance to the query the majority of approximations are filtered. For the remaining ones the corresponding exact data points have to be accessed. The critical factor of the VA-file' performance is the filtering step. If too many approximations remain, a lot of objects have to be accessed and the advantage of

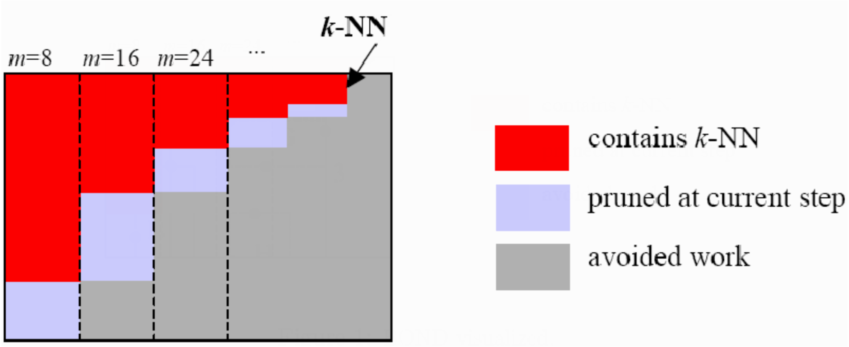


Figure 3.6: Example of BOND technique; figure taken from [AMNK02]

an approximation file is lost. Discussions and improvements of the VA-file pruning strategy have been reported in [WY06].

Also using a grid of cells, the IGrid [AY00] maintains separate lists for the objects in each cell. The similarity between any two objects uses only the set of dimensions for which the two objects lie in the same range (the *proximity set*). The number of objects that are accessed is kept small as the dimensionality increases, but the storage overhead is 100% because all the objects are copied into the indexing structure. Bitmap IGrid variants have been proposed [Cha03, GPB04].

The process of partitioning a high-dimensional space is itself a specific problem requiring dimension-wise discretization strategies. A discussion on discretizing continuous-valued attributes can be found in [FI92]. In [Jag06] dimension-wise histograms of the distance values from the points to the data center are created. The number of bins in each dimension's histogram gives the number of intervals. [AWY99] is also a data approximation technique where the partitioning strategy is based on a graph model.

3.4.6 What indexing method to use?

We have seen in this section that the high-dimensional indexing methods such as SAM and MAM divide the search space in a set of minimum bounding regions (MBR) hierarchically organized in tree structures. At search time, the MBR that don't overlap the query region are filtered. As dimensionality increases, the distances from a query object to the nearest and the farthest objects are almost in the same range. Practically, the nearest neighbor becomes indistinctive [KS01] from other neighbors, loosing its meaningfulness [Agg02, BGRS99]. In such conditions, the smallest query region that contains the nearest neighbor will overlap most of the MBR, making them non-prunable. The consequence is that the search methods end up accessing all the nodes of their structures in a pseudo-random fashion, which is more time consuming than a simple sequential scan —a problem often referred to as the "curse of dimensionality" [BGRS99, IM98].

To overcome this problems, in [KS01] the authors propose a distinctive-sensitive approach for tree-based structures. That is, to test the distinctiveness of the nearest neighbor in the course of search operation. Then, when it finds the nearest neighbor to be indistinctive, it quits the search and returns a partial result. Other indexing methods, such as the ones presented in Section 3.4.3, resort to single-dimension mapping. Another approach, presented in Section 3.4.4, uses specific aggregations with the goal of visiting only a fraction of the dataset. Finally, the data approximation methods sequentially access a compressed version of the data.

Choosing the proper index for a given situation should be the result of especially designed tests that consider various methods. A badly chosen indexing method may compromise efforts aimed at an effective use of the features and similarity measures. The behavior of the high-dimensional indexing methods depend on multiple parameters and data models. An indexing approach could behave well in some situations and worse in others. The metrics could be an example of a parameter that varies from method to method. For example, in [WY06] an important improvement over the original method in [WSB98] has been reported just by using a different metric.

To our knowledge, test frameworks for the whole range of high-dimensional indexing methods are not yet publicly available. The currently available frameworks, such as GIST [HNP95], SP-GIST [AI01] and XXL [dBBD$^+$01] provide only a subset of the existing indexing methods. GIST provides a tree-based template indexing structure. XXL, while also offering tree-based indexing templates, focuses on implementations of advanced database queries (cursors, iterators) independent from the underlying data types and data structures. As a consequence, many of the available approaches that cannot be tested in GIST or XXL are only compared to the sequential scan.

An infrastructure for the evaluation of the various indexing methods is essential to support the work at the higher levels in a complete retrieval chain. In Chapter 6 we propose an indexing method and a test framework for multidimensional search, allowing the configuration of various types of indexing methods, not only tree-based.

Chapter 4

The MetaMedia platform

This chapter introduces the MetaMedia retrieval platform, a solution for the management and search of multimedia databases. We present MetaMedia's architecture and functionalities that help us achieve such a goal. The platform integrates several components, namely a data model, an indexing method and an user interface. To illustrate the role of each component we instantiate the generic multimedia retrieval architecture presented in Chapter 2, showing that the data model and the indexing method belong to the database layer while the retrieval interface is at the user level. In between the database and the user layers, MetaMedia has a middleware layer with components for query analysis, answer computation, and metadata extraction.

Beside the architecture, this chapter also presents MetaMedia from the point of view of functionalities such as browsing, search, administration, content analysis and annotation.

4.1 Architecture

MetaMedia's three-layer architecture depicted in Figure 4.1 is as an instance of the generic architecture presented in Figure 2.1. The motivation for a layered architecture is that such an approach separates the components by their functionalities and demands well-defined interfaces between them. The system becomes more manageable, the parts can be developed independently and replacements are easier to make. To better understand each layer's role and the natural separation between them, is enough to image a simplified version of a retrieval system. The data storage component for example, can be modeled and built independently for simple data retrieval purposes. At the same time, a user interface component with browsing and simple search capabilities is required. In order to ensure the communication between the user interface and the database, a component in between, dedicated to query translation is required. A first separation can already be observed: the *database* and the *user* interface as stand-alone modules and the *middleware*, highly dependent on the first two. Placing new components in one of the *database*, *middleware* or *user* layers, is a matter of observing the data flow and their degree of independence. For example, an independent component that translates a natural language query into SQL queries would be generally placed between the *user* and the *database* layers, thus in the *middleware*.

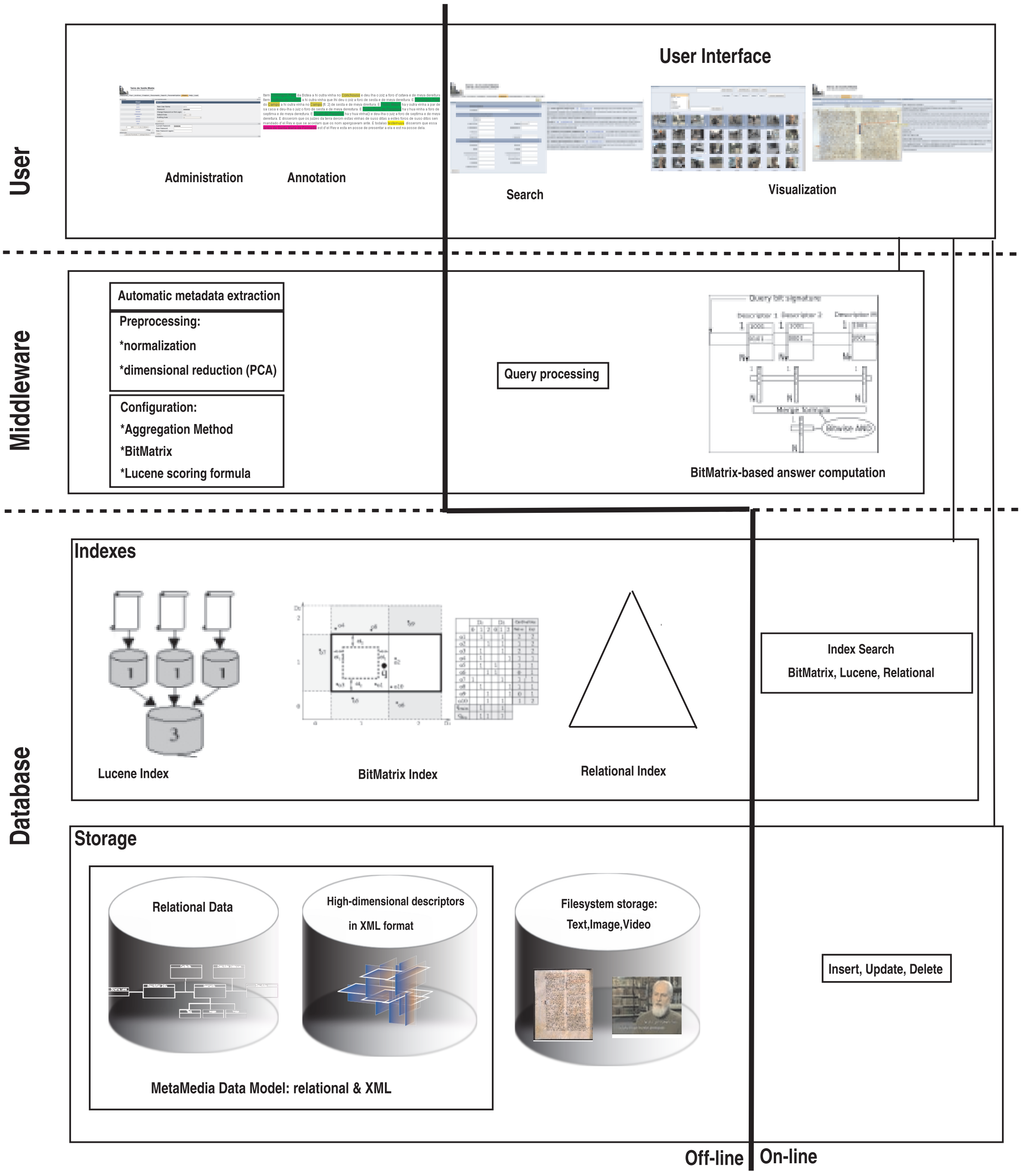


Figure 4.1: MetaMedia retrieval platform

MetaMedia started with a simple three-layer architecture, such as the one imagined earlier, that allowed the development of new components piecemeal. The availability of complex data such as the high-dimensional descriptors, and the plethora of standard presentation formats, brought advanced storage and search capabilities. This led to improvements in the data model and the development of specific indexing structures, both located in the *database* layer. Naturally, the *middleware* also became more consistent, hosting the business logic behind the answer computation, software for automatic content analysis and more complex query analysis.

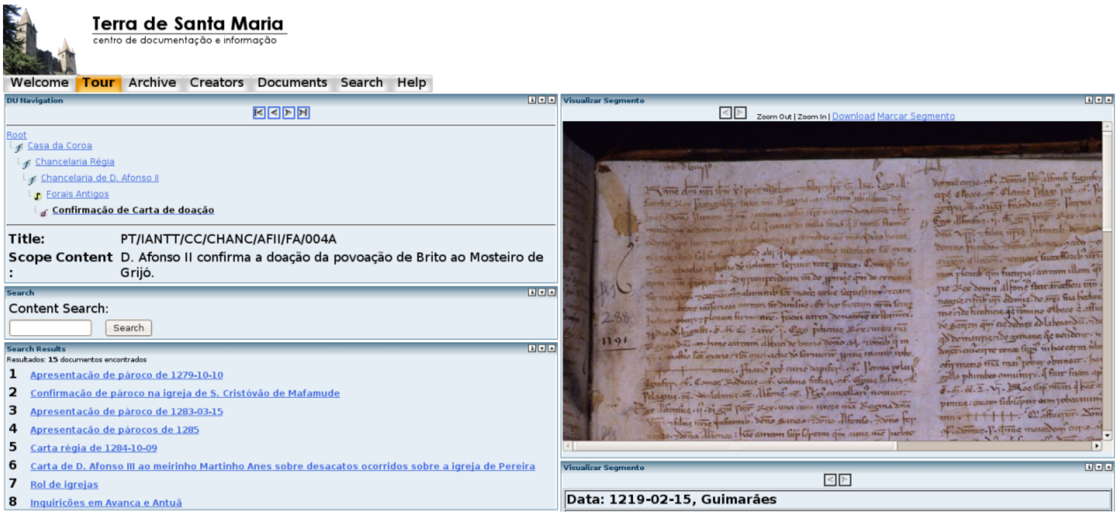
An important separation, marked by the vertical lines in Figure 4.1, is between off-line and on-line components. Components such as the content analysis, the index creation and configuration that are used for off-line tasks are separated at each level from components such as the query analysis and the index search that are used for on-line tasks.

The *database* layer is responsible for data storage and maintenance (creation, deletion, update) of the various index structures. At this level we distinguish between storage and indexing components. The storage components handle the input/output operations, such as uploads, deletes and updates for all the data and metadata types: text, image, video and high-dimensional XML descriptors. A first storage component is a hybrid relational-XML database containing all the metadata and the structuring information. Its organization is based on the MetaMedia data model, described in detail in Chapter 5. The second storage component is a file system in which all the raw content is gathered. Physically, the file system can be placed on a local disk or across network. The exact file locations are managed with the relational database.

Also parts of the *database* layer, the indexing components are redundant structures that speed up the search process. MetaMedia uses three types of indexes: BitMatrix high-dimensional indexes [CRD06] (detailed in Chapter 6), Lucene [Apa06] text indexes (detailed in Section 7.2) and relational indexes. The relational indexes are natively embedded in the database implementation that we use [Ora08] and are automatically created and consulted. The other two indexing structures are not native; we create them separately and also store them in the relational database. Figure 4.1 illustrates the *database* layer with its internal separation between storage and indexing; the available index types are shown as well.

The MetaMedia middleware layer accommodates software components dedicated to query analysis, similarity computation, and content analysis. The similarity computation and query analysis components, described in detail in Chapter 7, are responsible with the answer computation. For example, for a multimedia query that involves multiple descriptors, per-descriptor result lists are obtained by consulting BitMatrix indexes at database level and then aggregating them according to a configurable strategy. Several parameters such as the type of aggregation, the descriptor weights, or search parameters can be tuned at this level. Regarding the content analysis components, they are built on third-party software, namely the MPEG-7 reference software, also called MPEG-7 XM.

The user layer is represented by interfaces for collection management, query formulation, visualization of results and manual annotation. The MetaMedia user interface is built on a portal framework [Fou]. The framework makes use of reusable components, called portlets. We have



Root
Casa da Coroa
Chancelaria Régia
Chancelaria de D. Afonso II
Forais Antigos
Confirmação de Carta de doação

Title: PT/IANTT/CC/CHANC/AFII/FA/004A
Scope Content : D. Afonso II confirma a doação da povoação de Brito ao Mosteiro de Grijó.

Content Search:

Resultados: 15 documentos encontrados

1 Apresentação de pároco de 1279-10-10
2 Confirmação de pároco na igreja de S. Cristóvão de Mafamude
3 Apresentação de pároco de 1283-03-15
4 Apresentação de párocos de 1285
5 Carta régia de 1284-10-09
6 Carta de D. Afonso III ao meirinho Martinho Anes sobre desacatos ocorridos sobre a igreja de Pereira
7 Rol de igrejas
8 Inquirições em Avanca e Antuã

Data: 1219-02-15, Guimarães

Figure 4.2: Browse interface

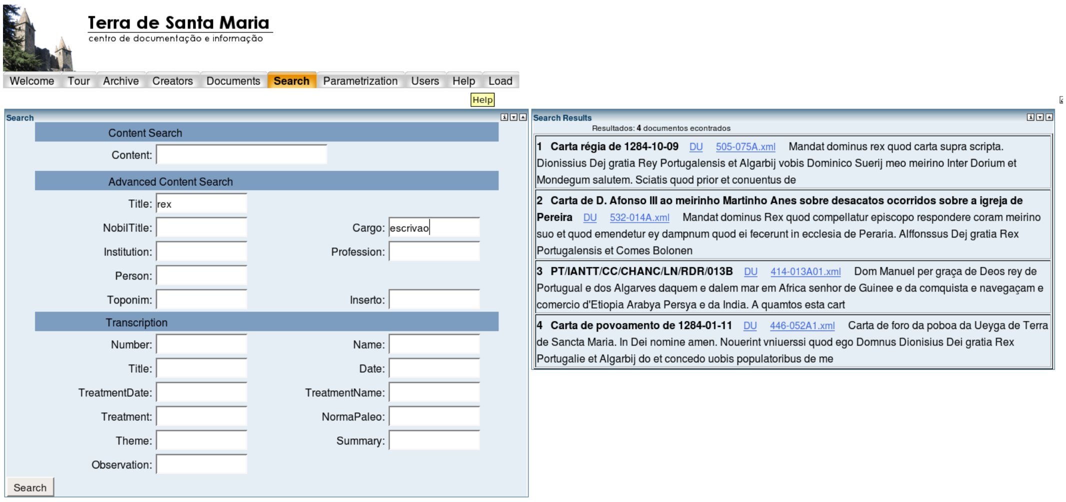
built such portlets for several search and visualization scenarios. Possible types of portlets include search portlets for experts and novice users, portlets that can display configurable degrees of detail or annotation portlets. For example, one of the portlets offers to novice users the possibility of taking a tour of the whole multimedia repository.

4.2 Functionalities

A multimedia retrieval system must support a large set of functionalities such as search, explore, administer, analyze and annotate. MetaMedia sums up the functionalities for the kind of applications we intend our framework to be used in. They are presented in the sequel.

4.2.1 Browser for multimedia repositories

Most of the current multimedia repositories can hardly be regarded as statical. The MetaMedia has been designed with a dynamic view of the repositories in mind. New multimedia items are often added and the existing ones are being updated. This can happen both with the content itself and the metadata. As a repository browser, MetaMedia uses components only from the User and the Database layers. Figure 4.2 is a screenshot of the browse interface used in the "Santa Maria da Feira" case study (see Chapter 8). On the left, from top to bottom, there are panels for navigation, simple text search and display of results. On the right part there are panels for the visualization of images and their transcriptions. The browse experience can be enhanced by maximizing or suppressing specific panels. For example, the parchment in the Figure 4.2 can be visualized in full-screen mode avoiding frequent use of the scroll bars.



Title: rex

Cargo: escrivao

Resultados: **4** documentos econtrados

1 **Carta régia de 1284-10-09** DU 505-075A.xml Mandat dominus rex quod carta supra scripta. Dionissius Dej gratia Rey Portugalensis et Algarbij vobis Dominico Suerij meo meirino Inter Dorium et Mondegum salutem. Sciatis quod prior et conuentus de

2 **Carta de D. Afonso III ao meirinho Martinho Anes sobre desacatos ocorridos sobre a igreja de Pereira** DU 532-014A.xml Mandat dominus Rex quod compellatur episcopo respondere coram meirino suo et quod emendetur ey dampnum quod ei fecerunt in ecclesia de Peraria. Alffonssus Dej gratia Rex Portugalensis et Comes Bolonen

3 **PT/IANTT/CC/CHANC/LN/RDR/013B** DU 414-013A01.xml Dom Manuel per graça de Deos rey de Portugual e dos Algarves daquem e dalem mar em Africa senhor de Guinee e da comquista e navegaçam e comercio d'Etiopia Arabya Persya e da India. A quamtos esta cart

4 **Carta de povoamento de 1284-01-11** DU 446-052A1.xml Carta de foro da poboa da Ueyga de Terra de Sancta Maria. In Dei nomine amen. Nouerint vniuerssi quod ego Domnus Dionisius Dei gratia Rex Portugalie et Algarbij do et concedo uobis populatoribus de me

Figure 4.3: The keyword search interface

4.2.2 Search

MetaMedia currently supports natural language queries, queries-by-keyword, structured queries, queries by low-level features and exploratory search. The queries by low-level features generally require efficient indexing of high-dimensional spaces. MetaMedia handles various dimensional spaces, arbitrary combination of features, specialized metrics and user-defined combinations of weights. For the search functionality, components from the three layers are involved. Two of the available search interfaces, namely the query-by-keyword and video search interfaces, are illustrated in Figures 4.3 and 4.4 respectively. Figure 4.3 is from the "Santa Maria da Feira" case study and illustrates a query-by-keyword. From the search fields in the left panel the "Title" and "Cargo" are filled with search terms and the results are displayed on the right panel. The interface in Figure 4.4 illustrates a video search session that took place in the context of a public video retrieval evaluation benchmark, called TRECVID; further details are given in Chapter 9.

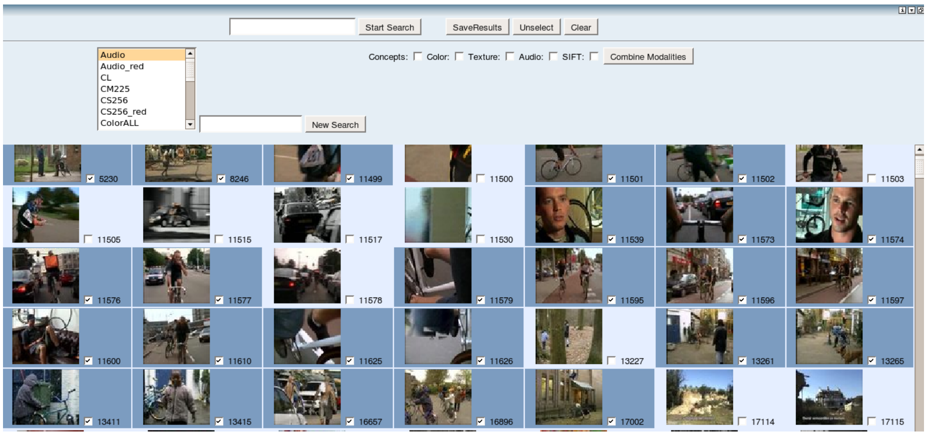


Figure 4.4: The video search Interface

4.2.3 Content Analysis

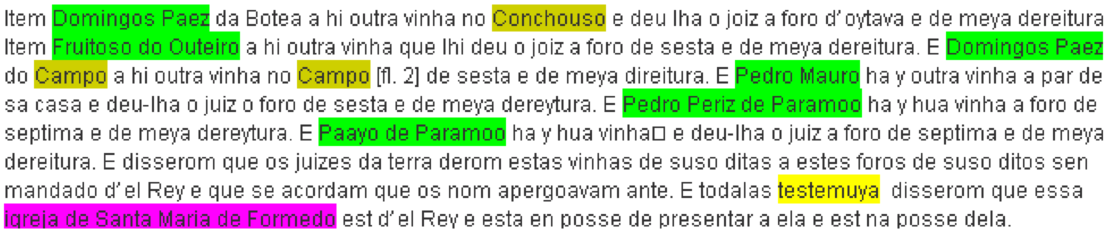
Dynamic addition of new items implies also integration of their automatically extracted content metadata in order to turn them searchable. If in the case of text items the content is naturally meaningful for the humans, the audiovisual content analysis produces low-level descriptors that lack a direct semantic link. Nevertheless, the low-level descriptors are useful for content-based similarity search, i.e query-by-example and the training of automatic concept detectors. MetaMedia offers content analysis for audiovisual data by incorporating third party software components. Their integration takes place at the level of *middleware* and *database* layers. An example of a DominantColor descriptor (see Section 3.1) obtained with the MPEG-7 XM tool is presented below.

```
<VisualDescriptor xsi:type="DominantColorType">
  <ColorSpacetype="RGB"/>
    <ColorQuantization>
       <Component>R</Component>
       <NumOfBins>8</NumOfBins>
       <Component>G</Component>
       <NumOfBins>8</NumOfBins>
       <Component>B</Component>
       <NumOfBins>8</NumOfBins>
    </ColorQuantization>
    <SpatialCoherency>31</SpatialCoherency>
    <Value>
       <Percentage>15</Percentage>
       <Index>255 255 255</Index>
       <ColorVariance>1 1 1</ColorVariance>
    </Value>
</VisualDescriptor>
```

4.2.4 Annotation

Manual annotation is the process of enriching the content with concepts that are considered relevant. Although are known to be subjective, error prone and expensive, the manual annotations are useful for datasets such as historic archives or for training automatic concept detectors. MetaMedia allows the manual annotation of multimedia items with concepts coming form concept sets such as locally controlled vocabularies, thesauri or ontologies. The choice of a specific concept set is dependent on the application domain. For example, in the video retrieval domain, the use of especially developed ontologies, namely LSCOM [Lar08, NST+06] and MediaMill [SWvG+06] have been considered more appropriate.

The annotation functionality requires components from the *user* and *database* layers. Figure 4.5 illustrates some highlighted regions of a fragment of a text document that was subject to the process of annotation. The collection of text documents from which the example in Figure 4.5 has been taken, was digitally preserved with MetaMedia and the annotation played an important role [RDB06]. We have included a detailed case study on this matter in Chapter 8. Some of the



Item Domingos Paez da Botea a hi outra vinha no Conchouso e deu lha o joiz a foro d’ oytava e de meya dereitura Item Fruitoso do Outeiro a hi outra vinha que lhi deu o joiz a foro de sesta e de meya dereitura. E Domingos Paez do Campo a hi outra vinha no Campo [fl. 2] de sesta e de meya direitura. E Pedro Mauro ha y outra vinha a par de sa casa e deu-lha o juiz o foro de sesta e de meya dereytura. E Pedro Periz de Paramoo ha y hua vinha a foro de septima e de meya dereytura. E Paayo de Paramoo ha y hua vinha□ e deu-lha o juiz a foro de septima e de meya dereitura. E disserom que os juizes da terra derom estas vinhas de suso ditas a estes foros de suso ditos sen mandado d’ el Rey e que se acordam que os nom apergoavam ante. E todalas testemuya disserom que essa igreja de Santa Maria de Formedo est d’ el Rey e esta en posse de presentar a ela e est na posse dela.

Figure 4.5: The annotation process

highlighted words in the figure are marked as *name*s, where the *name* concept comes from a locally defined controlled vocabulary. Similarly, other highlighted words were marked as *locations*.

4.2.5 Administration

The MetaMedia administration has two facets. One is the application administration which covers user management, maintenance and security issues. The other facet is the content administration. With respect to it there are three types of users, namely guests, editors, and archivists, which are granted different roles. For example, the guest users can browse and search the data but cannot insert or update it. The editors have enhanced roles, being allowed to edit context metadata or manually annotate contents. The archivists are the most privileged being, allowed to alter the structural organization of the data collections. For example, a task strictly reserved for the archivists is the design of the repository structure.

Chapter 5

The MetaMedia Data Model

The chapter describes the MetaMedia data model. Built with the goal of supporting current standards, the model accommodates content and context analysis metadata and provides the ground for their integration in search. We start by discussing some preliminary aspects concerning multimedia-oriented data models. The discussion is continued with a presentation of the main principles and from them the concepts that were identified in the metadata audiovisual standards and are underlying the proposed data model. Finally the data model is introduced.

5.1 Preliminary aspects

What data is there Multimedia databases display a large diversity both in the nature of stored objects and in the application domains. The database may be a repository of multimedia items which have been collected and appropriately described, as in a digital library; it may consist of the heterogeneous information assembled and dynamically modified in a Web site; or still result from the production process of a publisher or a broadcaster. In order to support such a diversity of applications, a multimedia data model must be designed for "any type of data", which currently means text, image, video, structured metadata (descriptive), unstructured metadata (content descriptors) and semi-unstructured data such as the annotations.

Focusing on metadata Standards such as Dublin Core can be applied to traditional descriptive metadata, MPEG-7 to content metadata and MPEG-21 to the various aspects of assembling components, handling digital rights, or adapting the content to specific players. The existing metadata standards help to clarify concepts and promote interoperability, but they are currently not appropriate as data models in a multimedia database [Bul04, ETM04]. There is no direct link between the metadata format, which is generally text/XML and the operationally effective representation required in practice for the data. A color histogram descriptor for example, is a high-dimensional vector embedded in XML format.

Among the aspects that have to be considered when building a metadata data model we emphasize here the need for integration of low- and high-level descriptors. The low-level descriptors, that correspond to features such as color, texture, shape or motion activity [DN04, MOVY01, Bob01, KIR+01], are radically different from the descriptive metadata such as “title”, “author”, “date”, also available for the objects. They are specific metadata in terms of representation, requiring specialized data structures for handling huge quantities of numeric values.

The recent manual and automatic annotation efforts tag the contents with high-level concepts that cover a wide range of semantics. This metadata category, called high-level content metadata, is different from the low-level content metadata or from the descriptive metadata. A multimedia data model must integrate the existing low-level descriptors with the high-level metadata [WvGC+05]. Queries such as “red cars”, or “portraits” imply both the low-level features such as color, texture or shape and the higher-level concepts such as “cars” or “faces”.

The model must also account for two kinds of separations. First, between the actual content and its description and second, between high- and low-resolution versions of the multimedia items. This model must also be able to incorporate materials from diverse sources, with various input formats and also to export them.

To fulfill this requirements, a clear identification of the main concepts underlying standards in different domains in required. The following sections introduce the central concepts of the MetaMedia model and the data model itself.

5.2 MetaMedia, from Principles to Concepts

The diversity of multimedia items requires the identification of abstractions that might be simultaneously useful for different kinds of objects and meaningful for the user when interfacing to the database.

The model is governed by three main principles. The first one is that multimedia objects are usually represented in a part-of hierarchical manner, allowing sets of items to be treated as objects that can have associated descriptions. This comes from the observation that multimedia items, have important semantics hidden in their tree-like structures. It is therefore essential to capture this kind of relationships in the model. There are other relations between items, but the part-of relationship is common and relatively easy to extend from the digital items themselves. It is also useful when exploring collections.

The second principle is that of uniform description, meaning that the same attributes are used for an individual object and for a set of related objects, allowing attributes to be inherited from a collection to a sub-collection down to individual items. This principle has been adopted in the standards for archival description such as ISAD [ISA99], ISAAR [ISA04] and EAD [EAD07] and proves itself very useful when it comes to the representation of large collections: metadata is frequently available for sets of items rather than individual ones, and inheritance can make it useful further down the hierarchy.

A hierarchy suited for one kind of collection may be of little use in a different one. It is therefore essential to offer flexible hierarchical structures with variable depths, and this is made possible by the principle of uniform description: introducing a new level in the hierarchy does not require the design of new descriptors. As an example, consider a collection of MPEG-21 items as

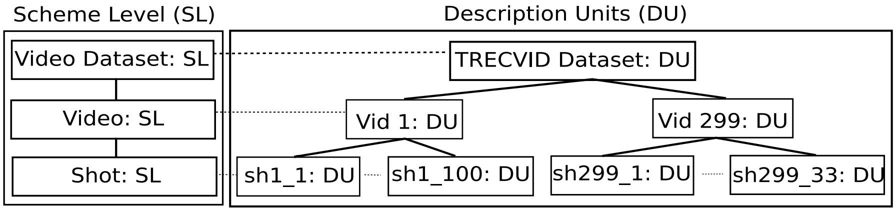


Figure 5.1: Simple Scheme Example

an item at collection level. If all the items in the collection share the same digital rights, those can be specified at collection level only.

The third principle identifies segments as content-only parts of multimedia items. They can be independently analyzed using content analysis techniques and specialized descriptors can be obtained for each of them. Segment can be given a very flexible representation, where descriptors coming from diverse feature extraction tools will fit. The definition of descriptors should be left as open-ended as possible. For example, the video track of a specific multimedia item may require motion activity and color descriptors, while the audio track of the same item may require a melody contour descriptor. Multimedia description standards such as [SKP02] have already identified a large set of such descriptors.

Following the principles of the model, three main concepts are used. The first one, the Description Unit (DU), corresponds to the concept of Description Units used in archival description [ISA99] and to Digital Items in the audiovisual standards. It captures the notion of an object or collection of objects that can be given a context in terms of creation and relationship with other objects. At the level of DUs we capture aspects such as creation context, terms of use and technical details. Such contextual information may become as important for the user as the content itself.

The second concept is the Segment, following the MPEG-7 vocabulary, capturing the notion of some part of a multimedia item that can be independently analyzed in terms of content and be manipulated independently, such as a part of a video sequence that is reused in a new documentary work. A segment has no context of its own, getting it from the DU of the object it belongs to.

In terms of structure both DUs and Segments are organized as part-of hierarchies. A DU is either a root unit or is a part of another DU, the collection where it belongs. The same applies to a Segment, which may be a part-of another Segment. Any item that can be individually retrieved has an associated context in its own DU.

The hierarchies of DUs can have various topologies and different semantics for their levels. Such hierarchies can be created for new collections or can be extracted from existing ones. In both cases they must capture the nature of the datasets. A Scheme that defines the possible levels, their semantics and their interconnections is required.

The mechanism that controls the structure of the hierarchies is incorporated in the model with the Scheme Level concept. Figure 5.1 illustrates a simple Scheme with only three Scheme Levels: *Video Dataset, Video* and *Shot*. Such a Scheme could belong to a test dataset, such as TRECVID, where a collection of videos is simply bundled and segmented. The right part of the figure, shows

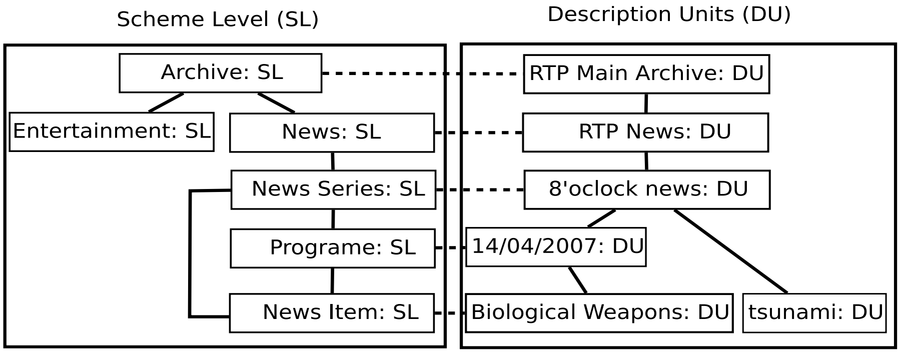


Figure 5.2: Scheme Instance Example

299 DU instances aligned with the *Video* Scheme Level that are part of the parent DU, “TRECVID Dataset”. Furthermore, DUs for each of the video shots are instantiated and linked to the corresponding parent DUs.

Figure 5.2 shows a different Scheme example and part-of hierarchy. This example shows an increased number of Scheme Levels and different ways of connecting them. On the left part we have the Scheme Levels, which in this example include *Archive*, *Entertainment* and *News*, *News Series*, *Programme*, and *News Item*. On the right, some DU instances are aligned with their respective levels. Thus, the DU called “8’oclock news” is a *News Series*, while the DU called “14/04/2007” is a *Programme*. The *News Item* on “Biological Weapons” is part of the “14/04/2007” *Programme* of the “8’clock news” *News Series*. After having specified an hierarchical model with the help of schemes, we can enable constraints related to it. For example, in Figure 5.2 the scheme allows *News Item* to be a child of *News Series*. Based on this permission, the specific DU “tsunami” (a *News Item*) can be directly linked to the “8’oclock news” DU (a *News Series*).

The third concept is that of a Descriptor. The sense in which Descriptor is used is the one established by the MPEG-7 standard—a representation of a feature [NL99]. In the model, a Descriptor is regarded as the result of a Segment analysis. An image Segment, for example, can be associated to its corresponding instance of the DominantColor descriptor, a video Segment can be associated to its MotionActivity descriptor and an audio Segment to its MelodyContour.

5.3 Data Model

Figure 5.3 illustrates the data model. The applicability of the MetaMedia concepts for a given dataset requires a previous analysis of the datasets’ nature and the usage scenarios. Upon analysis, the hierarchical structures are establishes as well as the amounts of uniform and specialized descriptions.

As an example, consider again the hierarchy of DUs in Figure 5.2. Attributes such as title, author, date and copyright are likely to apply both at a fine grain level such as the *News Item* Scheme Level and at a coarse grain level such as the *Archive* Scheme Level. If this is confirmed

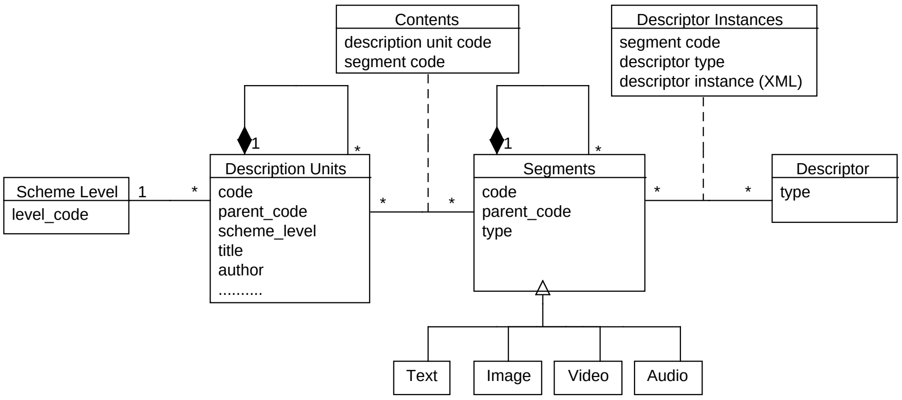


Figure 5.3: The MetaMedia Model

by the analysis, the attributes will be part of each DU's attribute set, thus uniformly present at all the levels.

The Scheme Level class is used to control the DUs hierarchy. For example in Figure 5.2 a DU that is an instance of a *News Item* Scheme Level can be a direct descendant of instances of *Programs* or *News Series* but not of instances of *News* or *Archive*.

The Segment class models independent content parts. The association between DUs and Segments, made through the Contents class, places each Segment in its own context —in a DU— and is interpreted as following: the Segments are the DUs' contents. The Segments can belong to several DUs at the same time and can be further categorized in text, image, video and audio types.

The Descriptor class models the specialized descriptors. The association Segment-Descriptors is modeled with the Descriptor Instance class. The descriptive information that is present at the level of Descriptors complements the description that exists in the DUs.

The proposed model was first tested in a prototype multimedia database [RDC04], which later became part of the MetaMedia's database layer [CAA^{+}07, RDC07a, RDC07b] (see Section 4.1). Its use has shown that different collections, with different usage scenarios, can be supported if the model offers flexibility in both the structure of the collection and metadata association.

Chapter 6

Data Indexing

In this chapter we present the BitMatrix index which has been our approach for multidimensional descriptor indexing. BitMatrix has been developed in the context of a broad high-dimensional indexes evaluation process which is also presented here.

It is generally recognized that automatically extracted descriptors are a cheap source of information on the media content, and current research focuses on exploring them in retrieval [LSDJ06, Moj04]. However such data is often high-dimensional in nature and difficult to index. Most of the indexing methods presented in Section 3.4 fail to handle datasets with hundreds or thousands of dimensions—the so called curse of dimensionality [BGRS99]. In this chapter, we will not be concerned with descriptors at semantic levels higher than the one of automatically processed features such as color or texture. We concentrate only on the specific task of retrieving objects that satisfy some (sharp) similarity criterion in a database of multidimensional descriptors.

One of the difficulties encountered in the evaluation of various high-dimensional indexing methods in general, and of the BitMatrix [CRD06] in particular, was the lack of a common platform on which they can be objectively tested. Indexing methods depend on parameters and storage models, which make them better suited for some situations and not as good in others. This makes them difficult to compare, and many of the proposed high-dimensional retrieval methods are only compared to the basic sequential scan. However, high-dimensional indexing methods follow common steps such as preprocessing the object data, partitioning the search space, index construction, query processing, searching the index, accessing the objects. We have developed a framework, called Multidimensional Multimedia Descriptor Indexing (MMDI), for the integration and benchmark of various indexing methods [GCRD07]. Indexing methods such as Sequential Scan, Bond [AMNK02], VA-File [WSB98], GridBitmap [Cha03], and the proposed BitMatrix have been included in the framework and compared.

The discussion is structured in four sections that cover two main aspects. In the first two sections the BitMatrix index is proposed along with an appropriate similarity criterion, while in the next two sections we present the MMDI evaluation framework and detail the BitMatrix results within it.

6.1 BitMatrix

Given the high cost of random disk access as compared to sequential access, the idea is to construct a collection of signatures that can be sequentially analyzed and used to effectively prune the search space. If such a structure is small enough to fit the main memory, than the I/O gain is of several orders of magnitude.

In the following, we assume a high-dimensional space with dimensionality N. We will also consider a dataset $\mathscr{O}$, containing $|\mathscr{O}|$ multimedia items (objects), each item being represented as a point is the high-dimensional space.

For each of the dimensions, the BitMatrix method follows a data approximation approach in the spirit of VA-File [WSB98] and IGrid [AY00]. Such an approach implies partitioning each of the N dimensions in k ranges.

A partition of a dimension D is a set of k_D ranges

$$\pi_D = \{r_i = [l_i, u_i], i = 1 \dots k_D\},$$

where l_i, u_i are the lower and upper bounds of range i.

Considering the possibility of having separate partitions for each dimension we define a *partitioning scheme*.

The partitioning scheme is the tuple $(k_1, k_2, \dots, k_N)$

The partitioning scheme is used to obtain bit *signatures* for all the objects in the dataset arranged as lines in a matrix.

Signature Given a partitioning scheme $(k_1, k_2, .., k_N)$, an *object's signature* is a bitmap of length $\Sigma_{D=1}^{N} k_D$. For each dimension the signature contains 1 for the range where the object belongs and 0 for the other ranges.

The bit signatures can be characterized by their *cardinality*.

The cardinality of a signature is the number of bits set to 1.

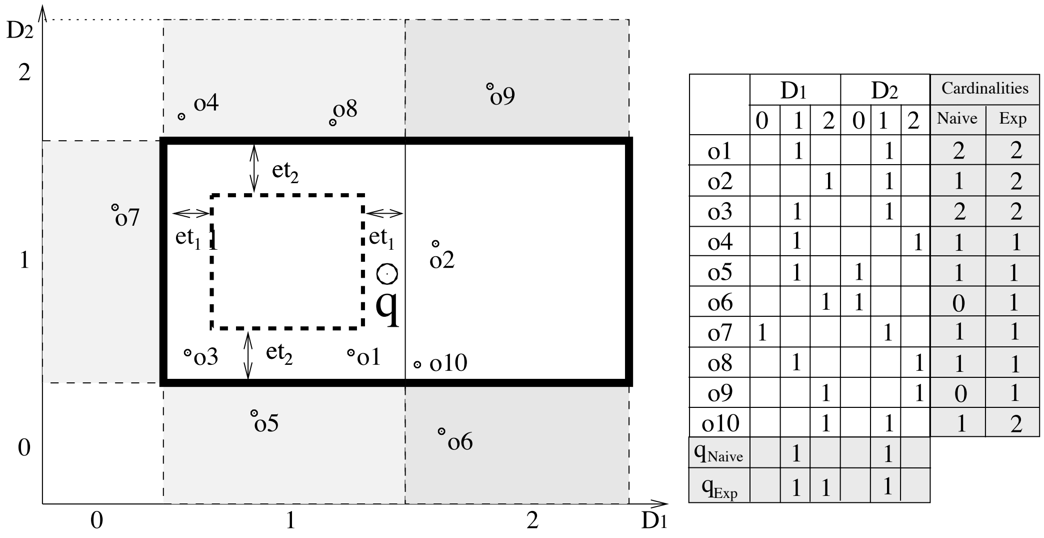
With the concepts of *partition*, *partitioning scheme*, *signature* and *cardinality* explained we can now see how a BitMatrix is effectively constructed.

6.1.1 Building the BitMatrix

The first step in the construction of the BitMatrix is to choose dimension-wise partitioning schemes such as *equi-width*, *equi-depth* or *k-means* partitioning. In the case of equi-width partitioning the ranges have the same length, while in the case of equi-depth the ranges contain an equal number of objects. The k-means partitioning requires a k-means clustering step. Because the k-means computation takes place dimension-wise, the k centroids can be sorted: $C_1 < C_2 < .. < C_k$.

The bounds of the k ranges are obtained from the centroids: the lower bound for range i is $l_i = (C_{i-1} + C_i)/2$ and its upper bound $u_i = (C_i + C_{i+1})/2$.

After partitioning all the dimensions we proceed to create the object signatures. As stated in the *signature*'s definition, for each object we check dimension-wise the ranges where it lies. Those ranges are assigned 1s and in the rest 0s. Note that the cardinalities of all the $|\mathscr{O}|$ objects are N, since every object signature has one and only one bit set to 1 per dimension.



	D1			D2			Cardinalities	
	0	1	2	0	1	2	Naive	Exp
o1		1			1		2	2
o2			1		1		1	2
o3		1			1		2	2
o4		1				1	1	1
o5		1		1			1	1
o6			1	1			0	1
o7	1				1		1	1
o8		1				1	1	1
o9			1			1	0	1
o10			1		1		1	2
q_{Naive}		1			1			
q_{Exp}		1	1		1			

Figure 6.1: BitMatrix

The example in Figure 6.1 has 10 objects in a two-dimensional space ($N = 2$) and 3 ranges per dimension ($k_1 = k_2 = 3$). On the right part of the figure, the corresponding BitMatrix structure is illustrated; for simplicity, only the 1s are shown for each object signature.

Object o_2 has signature 001010 as the object lies in range 2 for dimension 1, and in range 1 for dimension 2. Arranging each signature as a line in a matrix we obtain the BitMatrix with $|\mathscr{O}|$ lines and $\Sigma_{D=1}^{N} k_D$ columns.

6.2 Searching with the BitMatrix

We now propose two algorithms for approximate nearest neighbor, exploring the sequential access to the BitMatrix. The *naïve approach* selects objects based on the cardinality of the bitwise AND between object and query signatures, as follows.

Naïve search

- *Step 1*: Given a query object $q = [q_1, \ldots, q_N]$, obtain it's signature, i.e. find for each dimension D the range in which the query coordinate q_D lies.

- *Step 2*: Iterate through the objects, performing bitwise AND between their signatures and the query object's signature. If the cardinality of the resulting bitmap is above a predefined **cardinality threshold(ct)** the object is retained for the next phase.

- *Step 3*: Access the full vector values of the remaining objects, compute their exact distance to the query object and rank them.

We will use **cardinality of an object** in the sequel to refer to the cardinality of the bitmap resulting from the bitwise AND between the signatures for the object and the query. In Figure 6.1, the signature q_{naive}=010010 of the query object is AND'*ed* with the signatures of all the other objects $o_1, o_2, ..o_{10}$. With the cardinality threshold set to 2, only objects o_1 and o_3 remain for Step 3.

The example in Figure 6.1 shows that the naïve approach prunes object o_2 which happens to be the nearest neighbor. This effect, known as the *edge-effect* [AY00], appears because for all dimensions D, only the objects in the same range as q_D are considered. The **range expansion** heuristic is a modification of the naïve approach affecting Step 1: for a dimension in which the query object is close enough to one of the edges, the query's signature is set to 1 for both the query's range and the range next to it. Assuming that q_D lies in range i for dimension D, the expansion takes place to the left if $\|q_D - l_i\| < et_i \|u_i - l_i\|$ or to the right if $\|q_D - u_i\| < et_i \|u_i - l_i\|$, where et_i, the **expansion threshold**, takes values in $[0, 0.5]$.

Thus, the **range expansion search** is:

- *Step 1*: Given a query object $q = [q_1, \ldots, q_N]$, obtain it's expanded signature. That is, for each dimension D set to 1 the q_D's range. If $\|q_D - l_i\| < et_i \|u_i - l_i\|$ or $\|q_D - u_i\| < et_i \|u_i - l_i\|$ also set to 1 the range next to q_D.

- *Step 2*: Same as Step 2 in the naïve search.

- *Step 3*: Same as Step 3 in the naïve search.

Figure 6.1 also illustrates the effect of range expansion. The cardinalities column (with expansion) shows that with the query signature q_{exp}=011010 and the same cardinality threshold, objects o_2 and o_{10} are not pruned, as their cardinalities are now 2.

6.2.1 Subspace selection

The increase of dimensionality makes the task of finding the nearest neighbor harder because the distances between objects become very similar. For the majority of the high-dimensional search methods, the nearest neighbor becomes indistinguishable from the rest of the objects [BGRS99, HAK00]. In order to improve the quality of the nearest neighbor selection, we have considered the *subspace selection* approach, where subsets with smaller dimensionality are successively explored using the BitMatrix algorithms.

Let **s** be the number of dimensions of the subspace to be processed in the current iteration. The **subspace selection** search is:

- *Step 1*: Apply Step 1 and 2 of the naïve or expansion search approaches on the selected **s** dimensions ($\Sigma_{D=1}^{s} k_D$ columns of the BitMatrix).

- *Step 2*: Combine (intersection, union) the result set obtained from this subspace with the previous result set. If *the stop condition* is false repeat Step 1 on the next subspace.

- *Step* 3: Same as Step 3 of the naïve approach.

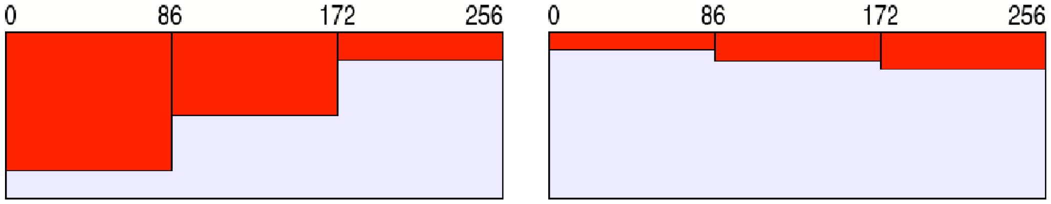


Figure 6.2: Subspace selection

The stop condition becomes true if enough dimensions have been processed or enough objects have been pruned. Figure 6.2 illustrates a space with total dimensionality 256 from which subspaces of s=86 dimensions are sequentially searched. The BitMatrix search is applied independently in each of the subspaces.

In the left part, a low cardinality threshold is used and intersection between the result sets is performed. Even though the low cardinality threshold acts like a low-pass filter, i.e large result sets remain, their intersection contains few objects. The objects remaining after intersection are illustrated in the dark region of each subspace, with the pruned ones in the rest. Thus, it can be observed that after searching the 0-86 subspace, a large set of objects remained. The search in 86-172 subspace produces a similar result (with respect to the size) set. However, after the intersection with the 0-86 result set, fewer remain as shown in the dark region of the 86-172 subspace. The final subspace, 172-256, produces by itself a relatively large results set, comparable in size to the 0-86 and the 86-172 ones. But after intersection, an even smaller set of objects remains.

In the right part of Figure 6.2, a high cardinality threshold is used and the union of the results sets is performed. Again, the dark regions show the remaining objects. In this case the cardinality threshold acts like a high-pass filter, thus few objects remain. The set of remaining objects increases with each subspace due to the union of the partial result sets.

The subspace selection feature is useful for speeding up the search in at least in two situations. First, when the dimensions of the original search space are not of equal importance. For example they may be ordered decreasingly by their importance and only a first subset of them used in the search. The second situation is when sufficient objects have been pruned after analyzing only some of the subspaces. The analysis of the remaining subspaces can thus be avoided and the search can stop earlier.

6.2.2 Insert, Update, Delete

The insertion of a new object in the BitMatrix, accounts for computing its signature and adding it as a new line in the matrix. The size of the BitMatrix grows linearly with the number of objects for a fixed dimensionality and linearly with the dimensionality for a fixed number of objects. The precise size of the BitMatrix is $(\Sigma_{D=1}^{N} k_D) * |\mathscr{O}|$ bits. To update an existing object its signature has to be modified with bitwise operations. To delete an object, the corresponding line is removed from the matrix.

6.3 Testing Multimedia Indexing Methods

The MMDI framework [GCRD07] that we propose is a step toward a systematic evaluation for the high-dimensional indexing methods. Each of the indexing methods presented in Section 3.4 brings specific implementation details and initial assumptions. They can only be tested in a flexible and robust framework. Several well-known methods have been already implemented in MMDI. The BitMatrix indexing method, also implemented on top of this framework, is compared to the other methods illustrating how new methods can be integrated and tested.

6.3.1 Application Requirements

Indexing methods are designed with efficiency in mind, and the associated data structures may become hard to use in some application domains. We have identified characteristics of the indexing methods which may not be widely taken into consideration, but which are central in our application domain. The MMDI framework is designed to allow testing for good performance in all the aspects below.

- Preprocessing: the amount of preprocessing work like normalization of the feature vectors and dimensional reduction, is an important factor when dealing with heterogeneous descriptors.

- Frequent updates: even if a substantial part of multimedia databases are regarded as statical in nature, as is the case with archives where append is the common operation, there are cases when updates are quite often. For instance, if the quality of some feature extraction tool improves, the results of applying it to an existing object will be incorporated as an update.

- Varying dimensionality: the dimensionality of our search space is directly related to the number of descriptors, meaning that each descriptor contributes with a number of dimensions. Adding a histogram descriptor, for instance, increases the dimensionality by the number of bins the histogram has been quantized (8, 16, 32, 166).

- Any dimensional subspace: at query time the user must be able to choose any subset of features.

- Support for different metrics: searching by different components of the feature space often implies using different metrics for each component. The aggregation may be done in different ways: arithmetic aggregate functions or fuzzy aggregate functions.
- Support for weighted queries: weights are used whenever the user want to stress the relevance of some features.
- Multiple query objects: the query may be specified as a composite of various query objects. In the case of relevance feedback, for example, the user selects multiple objects as positive/negative examples.

6.3.2 Framework Overview

The objective of the framework is to create a common platform for the evaluation of the high dimensional methods. The effort was mainly oriented towards the identification of the common operations of these methods that allow their integration in the framework.

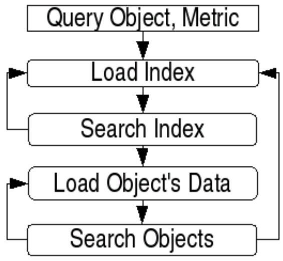


Figure 6.3: Framework overview

The framework structure emerged from the observed stages of the various retrieval methods. All methods require an initialization phase including normalization, space partitioning and index creation. The framework allows the definition of several types of partitions, such as equi-depth, equi-width and k-means. The search phase includes sub-tasks like loading the index (or fragments thereof), searching the index, loading the remaining object's data and searching them. The search flow for a generic method, illustrated in Figure 6.3, shows that tasks such as *Load Index*, *Search Index*, *Load Object's Data* and *Search Objects* allow variation in the way they are pipelined. For each indexing method only some of these paths will be required.

Evaluation using the framework can use facets like time, memory or disk space, quality of the retrieved objects and range of applicability. The comparisons can be done at method level or at specific sub-task level.

6.3.3 Framework Components

The concepts involved in the retrieval process are captured in the main components that will be described in the sequel.

Descriptor A descriptor represents an object's feature. It may contain information such as color histogram, dominant color or motion activity. It also specifies the number and the order of dimensions to be processed. A descriptor is captured as an object of class **Descriptor** with:

- descriptor id—the database identifier;
- name—the complete name;
- size—the number of dimensions;
- ranges—for each dimension, a set of ranges in which the data is to be partitioned;
- dimension order—the order by which descriptor dimensions are processed.

Query A query is captured as an object of class **Query** specifying one or more query objects, the descriptors that are used in the query process and the number of objects to be retrieved. The framework allows the definition of several query types, such as **nearest neighbor, k-nearest neighbors and range query**. Some of the query properties are optional depending on the query type and the method used to perform the retrieval.

Similarity Measure The similarity measure is used to analytically determine the similarity degree between a query object and a set of multimedia objects. The majority of the retrieval approaches use metrics (distance functions that satisfy the metric axioms), but the similarity measure is not necessarily a metric.

Similarity can be the result of grouping perceptual features, an action referred to in the literature as feature alignment [HK], or it can be estimated with models such as the Tversky's Feature Contrast Model [SJ97, Gen88].

The **Metric** abstract class in the framework allows the introduction of user-defined similarity formulas. When a multi- descriptor query is performed one must supply an aggregation method for combining the methods provided for each descriptor.

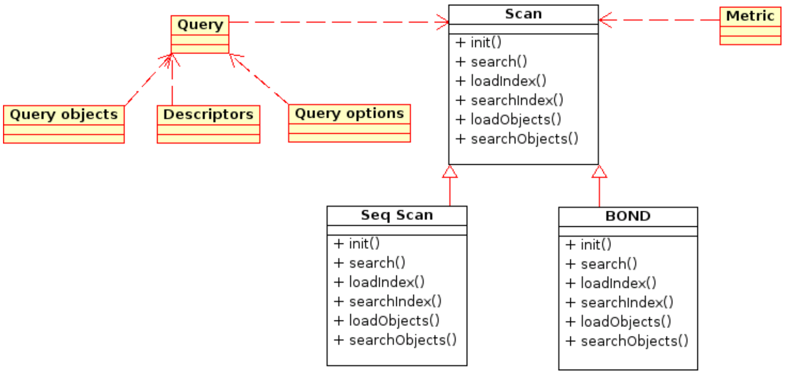


Figure 6.4: Scan method

Multimedia Object Objects are represented by the **MMObject** class. An object of this class stores a collection of values for each object descriptor. The same object structure is used throughout the retrieval process.

Scan Retrieval methods inherit from the **Scan** class and implement the abstracts methods named *init()* and *search(Query query, Metric metric)*, as shown in Figure 6.4. The search method receives a query where all options such as the query objects, the number of objects to retrieve and the query type are defined. A metric for retrieval is also supplied. In the retrieval process the indexing technique is used to prune the search space according to the specified options. After the pruning stage a sequential scan is performed to sort the remaining objects. The framework currently supports the following methods:

- Sequential Scan,
- Branch-and-bound ON Decomposed Data (BOND)[AMNK02],
- VA-File [WSB98],
- Grid Bitmap [Cha03],
- BitMatrix [CRD06].

ScanTest The **ScanTest** provided in the framework is a class developed on top of the JUnit [JUn05] testing library and is the part of the framework where tests are configured and results are collected.

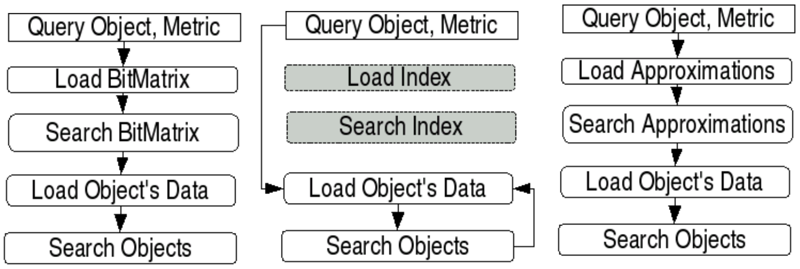
6.3.4 Technological platform

MMDI is implemented in Java taking advantage of portability and the richness of its associated technologies. As XML-based standards like MPEG-7 [SKP02] are the most common languages for the representation of the descriptors, the framework assumes XML as its input format. Nevertheless, user-defined methods for acquiring data in different formats can be easily integrated. XMLBeans [Apa05] is used for dealing with XML inputs, Weka [WF05] is used for the clustering facilities, and COLT [Ope05] for the base implementations of the **Bitmatrix** and **Bitvector** classes.

6.3.5 Integrating Indexing Methods

For the BOND [AMNK02], VA-File [WSB98] and Grid Bitmap [Cha03] methods, the basic operations for loading and searching the index, as well as loading and searching the objects, have been identified and adapted to the common platform.

BOND works directly on the original data, i.e. it does not use an index structure. It loads dimensions one by one and, after processing a few dimensions, the objects that cannot further participate in the result are discarded. The iterative application of this procedure leads to a small



(a) The BitMatrix (b) BOND (c) VA-File

Figure 6.5: Search flows for various retrieval methods

remaining subset of objects which are exhaustively scanned. The only framework tasks that were identified in BOND were therefore *loadObjects's Data* and *searchObjects*.

A similar analysis on VA-File and Grid Bitmap methods show that both have the complete chain: *loadIndex*, *searchIndex*, *loadObjects's Data* and *searchObjects*.

6.3.6 The BitMatrix in MMDI

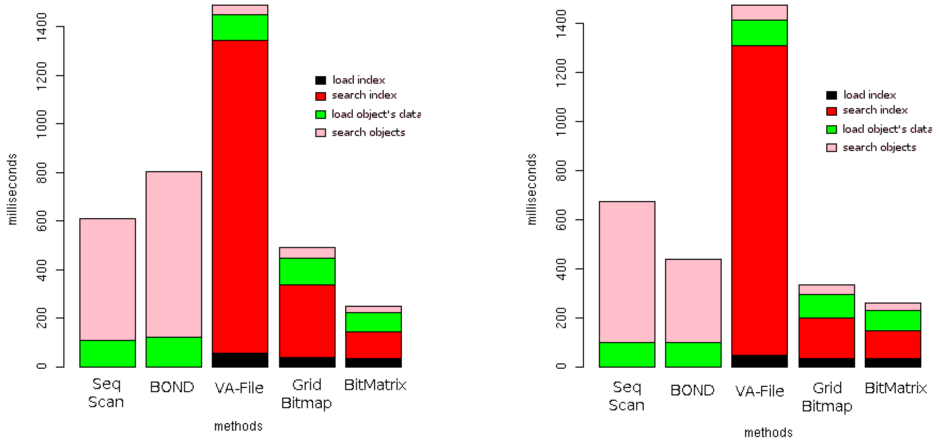
To give more insight on how new indexing methods can be used in the framework, we will now use the BitMatrix [CRD06], the method we have presented in Section 6.1. It has been said that the BitMatrix method follows a data approximation approach partitioning each dimension in ranges. The partitioning scheme is then used to obtain bitmap signatures for all the objects. The search algorithm consists of the bitwise ANDs between object and query signatures. Figure 6.1 illustrates a two-dimensional space having each dimension partitioned in three ranges.

To integrate this method in MMDI, a new **BitMatrix** class is derived from the **Scan** class, inheriting the general behavior from its parent. The set of methods in the newly created BitMatrix class is the same as in the BOND or Sequential Scan; the methods are illustrated in Figure 6.4. The *init()* method, through the helper classes **DBInserter** and **RangeAnalyser**, takes care of pre-processing, space partitioning and index construction. The partitioning method can be equi-width, equi-depth or k-means. For the k-means partitioning, a previous k-means clustering is performed using the Weka library [WF05]. The resulting BitMatrix structure has been implemented with the help of the COLT library [Ope05], a library that is prepared for bit operations by having implemented dedicated classes such as the BitVector.

The integration also requires the implemention of the *search(Query q, Metric m)* method for the **BitMatrix** class and overloading as needed the *loadIndex()*, *searchIndex(Query q,Metric m)*, *loadObjects(Collection objectIDs)* and *searchObjects(Query q, Metric m)* methods.

6.3.7 Search Flows in MMDI

Figure 6.5 illustrates the search flows of some of the methods that are already integrated in the framework. The search flow for the BitMatrix loads the index, i.e. the BitMatrix, searches the



(a) Manhattan metric

(b) Euclidian metric

Figure 6.6: Varying the metric on i300

index (Search BitMatrix) and then loads and sorts the remaining objects. The search flow for BOND includes only the *loadObjects* and the *searchObjects* methods. The shaded boxes indicate that in fact BOND does not use an index. The VA-File loads the index, i.e the approximations file, then sequentially analyzes all the approximations; finally the remaining objects are loaded and sorted.

In order to test the newly integrated BitMatrix, a **BitMatrixTest** class derived from the **ScanTest** class was implemented in order to accommodate test scenarios with several parameters: number of iterations, number of objects, number of distance computations, number of dimensions processed at each step and several threshold values.

6.3.8 Method comparisons with MMDI

In this section we show the kind of results that can be obtained with the MMDI framework. From the set of indexing methods presented in Section 3.4, the SAM's, MAM's and single-dimension mapping were not implemented in the framework. This is due to a set of initial requirements that we have identified and which the above mentioned methods do not satisfy: we only keep the high-dimensional indexing methods that retain most of the sequential scan's flexibility with as few assumptions as possible on the metric, dimensionality and structure of descriptors. However, their integration in our framework would have been straightforward as well; such methods have already been implemented in frameworks like GIST [HNP95] and XXL [dBBD^{+}01].

The experiments reported were run over two datasets. The first dataset, i300, is a small dataset of 300 images with 256-dimensional histograms and 3 ranges per dimension. The second, i9000, consists of 9908 image with 256-dimensional histograms and 7 ranges per dimension.

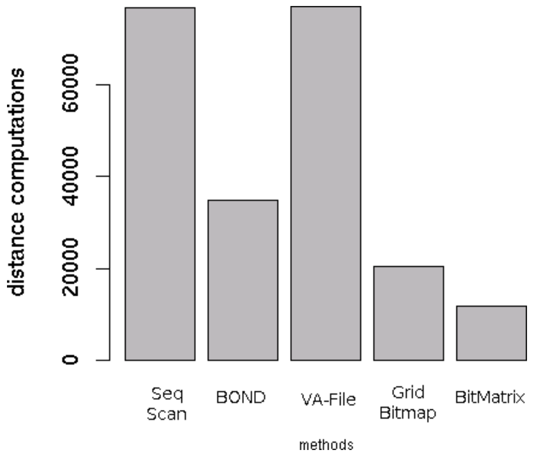


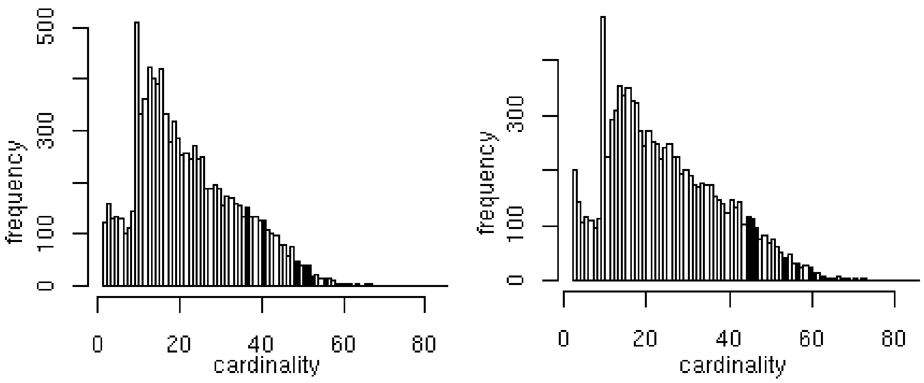
Figure 6.7: Distance computations; i9000 dataset

The time results in Figure 6.6 illustrate the search times as sums of the times for the various sub-tasks: load index, search index, load object's data, search objects. Figure 6.6 shows the time results for the i300 dataset, using the Manhattan and Euclidian metrics. The time for the VA-File does not do justice to the method as it is obtained using the same partitioning scheme as GridBitmap and BitMatrix. For the case of i300, the partitioning scheme makes 3 ranges in each of the 256 dimensions, thus 3^{256} cells. With the non-empty cells containing at most one object the pruning efficiency of the VA-File (search index time in Figure 6.6) is severely reduced. Sequential Scan and BOND load the whole dataset sequentially in memory, while the other methods load (pseudo- randomly) a much smaller subset representing the objects that are not pruned. Even so, in the case of i300 the time values for the load object's data task are very similar, confirming the advantage of the sequential access over the random access.

It is also interesting to study the effects of metric change. In Figure 6.6 (a) the Manhattan metric is used while Figure 6.6 (b) shows the same results for the Euclidean metric. It can be observed that with the Euclidean metric the search time for BOND decreases, leading to a better general score.

The number of distance computations performed by each of the evaluated methods is another result that can be obtained with the MMDI. For this purpose, the i9000 dataset has been used. The statistics of distance computations are illustrated in Figure 6.7. It can be observed that the Sequential Scan and the VA-File perform distance computations for each object, i.e 9908 distance computations, while the other methods compute the exact distances only for the set of objects remaining after their pruning phases.

The implementation of the MMDI framework has shown that changes in the partitioning system and in the metrics can significantly influence the performance of a method. Rather than choosing a single indexing method, the goal of MMDI is to observe the behavior of the multidimensional indexing methods at a fine-grained level.



(a) No Expansion

(b) Exp threshold=0.01

Figure 6.8: Histogram of cardinalities

6.4 Evaluation of the BitMatrix Method

The BitMatrix is an approximate search method and trades precision for speed by means of parameters such as cardinality threshold, range expansion threshold, partitioning scheme and subspace selection, as presented in Section 6.1. Note that BitMatrix deals with objects represented as bit signatures. Such signatures are derived from partitioning schemes, i.e. each dimension is divided into intervals (ranges). For the ranges in which the objects are located, the bit signatures contain 1s, and 0s otherwise. The similarity between two objects is given by the cardinality of the bitwise AND between their signatures, where the cardinality represents the number of bits set to 1. In other words, this similarity measure counts the number of dimensions where the objects coincide (share the same range). In some dimensions however, two objects can be close to each other but located in different ranges, which means they are both close to the common edge. This "edge-effect" depends on the object distribution and the dimension partitioning method.

The BitMatrix evaluation has been performed on two datasets: the dataset mentioned earlier of 9908 real images with 256-dimensional histograms —the i9000 dataset and a synthetic dataset, i10000, of 10000 objects uniform independent identical distributed (IID) in all of its 80 dimensions.

6.4.1 Range expansions evaluation

To illustrate the effect of range expansions in *k-nearest neighbors* search $NN_k(q)$, we consider a subspace of 86 dimensions from the original 256-dimensional i9000. The histograms of cardinalities in Figure 6.8 show the cardinalities (horizontal-axis) and the number of objects with the same cardinality (the frequency axis): on the left-hand side without expansion and on the right with an expansion threshold $et = 0.01$. The cardinalities (the bins) of the first 10 nearest neighbors (NN_{10}) are illustrated as dark bins in both the left-hand and right-hand histograms. It can be observed that after expansion, all the 10 nearest neighbors have cardinalities larger than 40.

	i9000 $N = 256, k_D = 7, i = 1 \ldots 256\,(k - means\,partitioning)$							
ct	$NN(q)$				$NN_{10}(q)$			
	Naïve(et=0)		**et=0,01**		Naïve(et=0)		**et=0,01**	
	Recall	accessed	Recall	accessed	Recall	accessed	Recall	accessed
0.73	0.6	0.2%	0.7	0.4%	0.39	0.2%	0.48	0.4%
0.67	0.78	0.8%	0.87	1.0%	0.61	0.8%	0.68	1.0%
0.55	0.93	2.9%	0.95	3.8%	0.86	2.9%	0.9	3.8%
0.47	**0.97**	**6.0%**	**0.99**	**7.4%**	**0.91**	**6.0%**	**0.94**	**7.4%**
0.40	**1.0**	**10.9%**	**1.0**	**13.5%**	**0.93**	**10.9%**	**0.96**	**13.5%**
	i10000 $N = 80, k_i = 7, i = 1 \ldots 80\,(k - means\,partitioning)$							
	$NN(q)$				$NN_{10}(q)$			
	Naïve (et=0)		**et=0,01**		Naïve (et=0)		**et=0,01**	
0.60	0.24	0.19%	0.25	0.21%	0.08	0.19%	0.09	0.21%
0.50	0.55	1.99%	0.57	2.31%	0.33	1.99%	0.35	2.31%
0.40	0.78	8.16%	0.84	9.32%	0.57	8.16%	0.62	9.32%
0.35	0.90	17.0%	0.93	19.2%	0.77	17.0%	0.80	19.2%
0.30	0.90	32.2%	0.94	34.21%	0.85	32.2%	0.87	34.21%

Table 6.1: Testing the BitMatrix on two datasets

6.4.2 Recall Performance

The next set of experiments was geared towards observing the BitMatrix performance for nearest neighbor search ($NN(q)$) and k-nearest neighbors search ($NN_k(q)$). To illustrate the $NN_k(q)$ search performance, we compare the approximate BitMatrix results with the exact *k-nearest neighbors*. We use the recall rate for this purpose.

Assuming R as the $NN_k(q)$ set and A as the set of approximate neighbors obtained with BitMatrix, the recall rate is

$$\frac{|A \cap R|}{|R|}.$$

This measure has been proposed to evaluate retrieval in a context where user relevance judgments are available. Here it is used in a more formal evaluation. Therefore we assume that the relevant objects (the R set) are only the first k neighbors obtained with a sharp similarity criterion, such as the euclidean metric.

Beside the recall rate scores, we also record the percentage of objects that are effectively accessed, i.e that remain after the pruning phase. This is expressed as the ratio of $|A|$ to the size of the dataset.

Table 6.1 shows average values of the two measures (recall rate, and % of objects accessed) with respect to $NN(q)$ and $NN_{10}(q)$ across random sets of 100 queries. The cardinality thresholds are in the first column. With a cardinality threshold of 0.55 for instance, less than 3% of i9000 is accessed, the average recall rate is 0.93 (relative to NN) and 0.86 (relative to NN_{10}). The experiments have shown that the trade-off between quality of retrieval and speed can be tuned with the expansion mechanism. For example, with the cardinality threshold 0.47, about 6% of i9000 is accessed, and the recall rate relative to NN_{10} is 0.91. If range expansion is performed the NN_{10} recall becomes 0.94 at 7.4% accessed objects, while for a smaller cardinality threshold (ct = 0.4) the NN_{10} recall is 0.93 at 10.9% accessed objects. With an increase in recall (0.94 vs 0.93) and less 3.5% (7.4% vs 10.9%) objects accessed, expansion should be preferred in this case.

The results for the synthetic uniform IID distributed i10000 are presented in the second half of Table 6.1. The numbers indicate lower performance than on i9000. Much larger amounts of the i10000 have to be analyzed in order to obtain acceptable recall rates. Note however, that i10000 is a synthetic dataset independent and identically uniform-distributed across all the dimensions, thus not a realistic one. The expansion mechanism clearly improves the recall rate in this case as well.

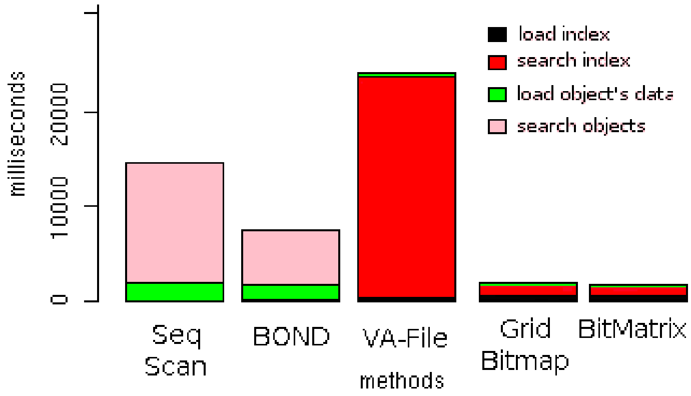


Figure 6.9: Comparing methods that use the euclidean metric

6.4.3 Time performance

Another set of experiments have been designed to observe the time performance of the BitMatrix as a memory-based indexing method. They have been performed on i9000. The time columns in Figure 6.9 have four components: the time to load the index in memory (if such an index exists), the time to search it, the time to load objects data, and the time to search them. The total time for the BitMatrix is clearly smaller than the values for Sequential Scan and Bond [AMNK02] and is in the same range as the GridBitmap [Cha03]. The VA-file's [WSB98] time again is not favorable to the method as it is tested using the same partitioning scheme as GridBitmap and BitMatrix; with 7 ranges in each of the 256 dimensions, there are 7^{256} approximation cells, with the non-empty ones having at most one object. Thus, VA-file has to access much more cells than real objects.

Chapter 7

Retrieval Method

In this chapter the discussion is focused on MetaMedia's search and retrieval processes. The complex nature of the multimedia items lead to the development of several search approaches that target different parts of them. MetaMedia integrates search on textual content, search on structured contextual metadata and search on visual features for image and video segments; each of them is presented independently in the sequel. Then, the retrieval method will be described as faceted way of integrating several search engines in a powerful and flexible user interface.

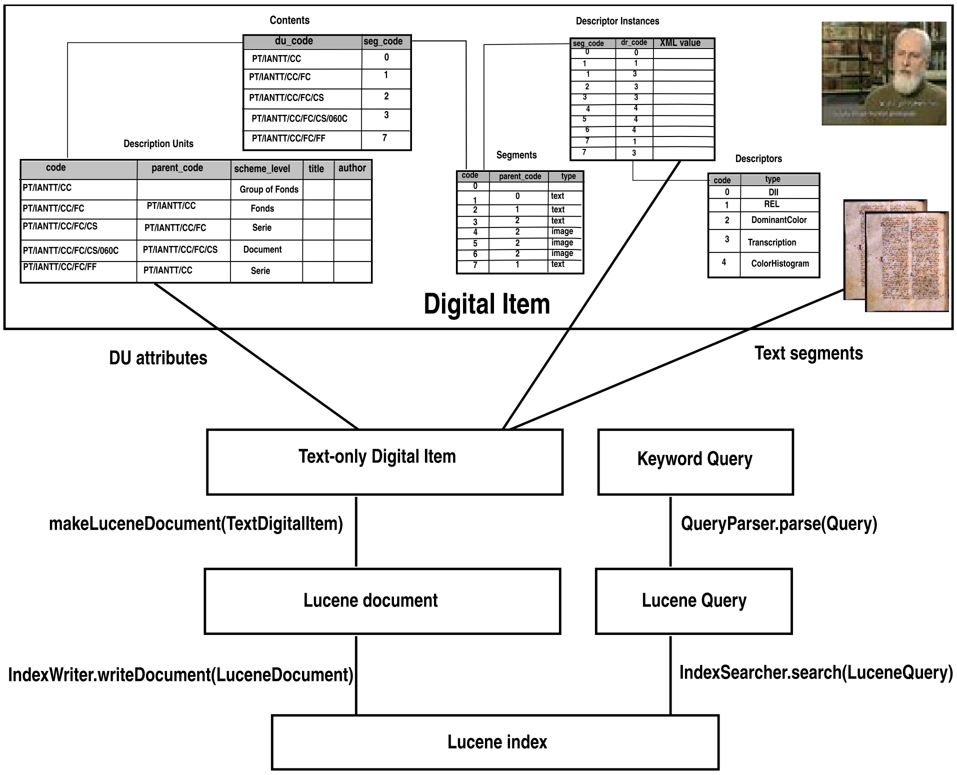
7.1 Search Modalities

Taking advantage of the data model, MetaMedia proposes a hybrid search approach where queries can be formulated in several ways:

- Query by keyword is the most straightforward mode, allowing the exploration of the collection by users with no special skills.
- Structured queries are aimed at more specialized users, who have some background on the domain covered by the contextual descriptors and understand the structure of the collection.
- Query by content uses audio-visual features. It is essential when items are mostly images and videos and the users want to provide example items or express image features.
- Exploratory search allows users to browse the entire collection in a navigation mode.

All these search modalities will be detailed in the following sections. However, at this point, two general observations can be made. The first one concerns the relationship between search modalities and index types. Each search modality relies on dedicated indexing structures that are targeted for specific types of data. Thus, the keyword-based search uses text indexes, the structured and exploratory search modalities use relational indexes, and the content-based search uses high-dimensional data indexes.

A second observation concerns the relationship between the index types and the data model. On one side, the fact that the indexes can store only certain types of data, forces a separation of the



Contents

du_code	seg_code
PT/IANTT/CC	0
PT/IANTT/CC/FC	1
PT/IANTT/CC/FC/CS	2
PT/IANTT/CC/FC/CS/060C	3
PT/IANTT/CC/FC/FF	7

Description Units

code	parent_code	scheme_level	title	author
PT/IANTT/CC		Group of Fonds		
PT/IANTT/CC/FC	PT/IANTT/CC	Fonds		
PT/IANTT/CC/FC/CS	PT/IANTT/CC/FC	Serie		
PT/IANTT/CC/FC/CS/060C	PT/IANTT/CC/FC/CS	Document		
PT/IANTT/CC/FC/FF	PT/IANTT/CC	Serie		

Descriptor Instances

seg_code	dr_code	XML value
0	0	
1	1	
1	3	
2	3	
3	3	
4	4	
5	4	
6	4	
7	1	
7	3	

Segments

code	parent_code	type
0		
1	0	text
2	1	text
3	2	text
4	2	image
5	2	image
6	2	image
7	1	text

Descriptors

code	type
0	DII
1	REL
2	DominantColor
3	Transcription
4	ColorHistogram

Figure 7.1: Text Indexing & Keyword Queries

searchable information into text, relational, image, video and high-dimensional data. On the other side, the data model described in Chapter 5, is not datatype-oriented. The information's place in the model is acquired based on the relationship with the digital items. There can be content parts —the Segments– of any modality, for example. Similarly, there are no datatype restrictions for Description Units or Descriptors. The conceptual separation induced by the data model is orthogonal to the datatype-centric separation that is required for indexing. It is therefore natural to simultaneously index, for example text information, from Segments, DUs, Descriptors, or any other part of the model.

7.2 Query by keyword

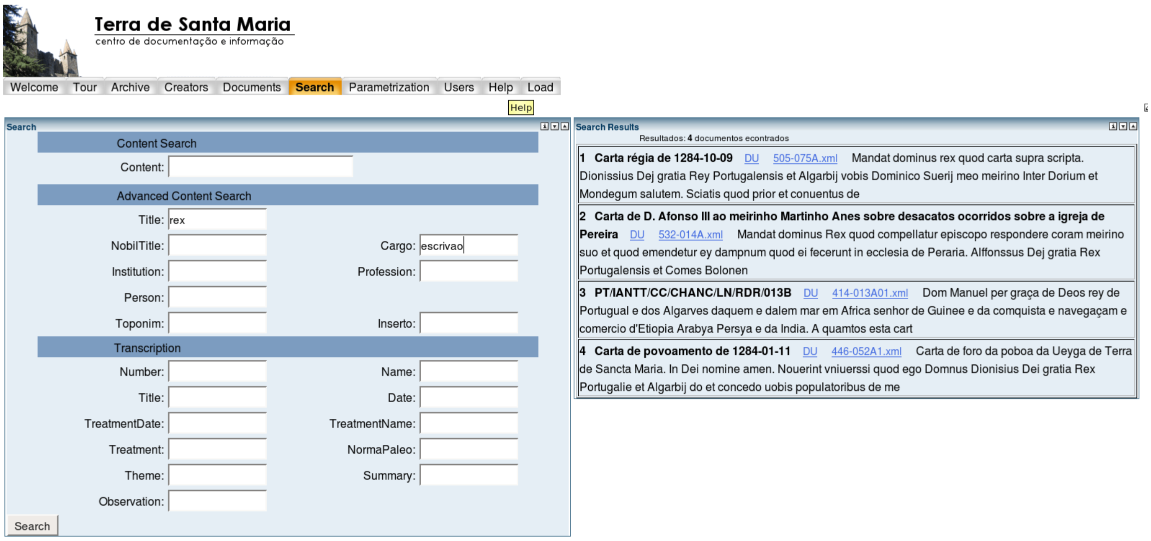
The first search modality is query-by-keyword; it requires a full-text indexing system. The Lucene engine [Apa06] has been used to index all the textual information of the digital items which, according to our data model, is located in Segments (text segments), DUs (descriptive information) and Descriptors. The Lucene indexing system works internally with so-called *Lucene documents*, where a *Lucene document* is a collection of *fields*. A *field* is a pair composed of a name and value: $< name, value >$. The indexing and search operations are mostly taken care by the Lucene programming interface through its IndexWriter, QueryParser and IndexSearcher classes. The user's only responsibility is to construct valid Lucene documents from the available text information.

```
public Document makeLuceneDocument(TextDigitalItem tdi){
//new Lucene Document
    Document doc = new Document();
//fields from DU
    doc.add(new Field(``path'', tdi.path));
    doc.add(new Field(``DU_code'', tdi.DU_code));

//fields from the text segment
    doc.add(new Field(``título'', tdi.title));
    doc.add(new Field(``cargo'', tdi.cargo));
    .
    .
    doc.add(new Field("content", tdi.content));
    return doc;
  }
```

Figure 7.2: Building Document representation in Lucene

Figure 7.2 illustrates the process of creating one Lucene document from a digital item. The fact that the function *makeLuceneDocument* accepts a “TextDigitalItem” parameter, suggests that only text data from the digital items is indexed. The upper part of Figure 7.1 shows that in general, such text data can be located in every part of the data model such as the Segments, the DUs or Descriptors. In practice however, the main sources of text information are the DUs and the Segments. Figure 7.1 also illustrates that the Lucene documents are written to the index file with the help of the Lucene's IndexWriter class.



Content Search

Content:

Advanced Content Search

Title: rex

NobilTitle:

Cargo: escrivao

Institution:

Profession:

Person:

Toponim:

Inserto:

Transcription

Number:

Name:

Title:

Date:

TreatmentDate:

TreatmentName:

Treatment:

NormaPaleo:

Theme:

Summary:

Observation:

Search

Resultados: **4** documentos econtrados

1 **Carta régia de 1284-10-09** DU 505-075A.xml Mandat dominus rex quod carta supra scripta. Dionissius Dej gratia Rey Portugalensis et Algarbij vobis Dominico Suerij meo meirino Inter Dorium et Mondegum salutem. Sciatis quod prior et conuentus de

2 **Carta de D. Afonso III ao meirinho Martinho Anes sobre desacatos ocorridos sobre a igreja de Pereira** DU 532-014A.xml Mandat dominus Rex quod compellatur episcopo respondere coram meirino suo et quod emendetur ey dampnum quod ei fecerunt in ecclesia de Peraria. Alffonssus Dej gratia Rex Portugalensis et Comes Bolonen

3 **PT/IANTT/CC/CHANC/LN/RDR/013B** DU 414-013A01.xml Dom Manuel per graça de Deos rey de Portugual e dos Algarves daquem e dalem mar em Africa senhor de Guinee e da comquista e navegaçam e comercio d'Etiopia Arabya Persya e da India. A quamtos esta cart

4 **Carta de povoamento de 1284-01-11** DU 446-052A1.xml Carta de foro da poboa da Ueyga de Terra de Sancta Maria. In Dei nomine amen. Nouerint vniuerssi quod ego Domnus Dionisius Dei gratia Rex Portugalie et Algarbij do et concedo uobis populatoribus de me

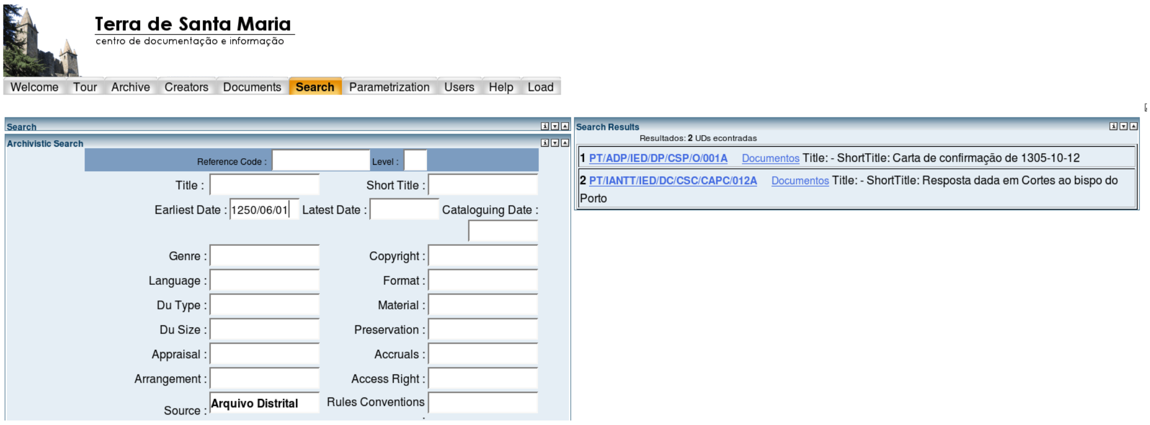
Figure 7.3: The keyword search interface

The search process, schematically illustrated in Figure 7.1, starts with a keyword-based query introduced in the dedicated user interface that is shown in Figure 7.3. The interface has been constructed in the context of a case study, called Santa Maria da Feira, which will be detailed in

Chapter 8. The keyword queries are passed from this interface to the Lucene QueryParser which in turn issues Lucene queries. For example, a query for documents containing the term *titulo* of value “rex” or the term *cargo* of value “escrivao”, is put by the QueryParser into the following Lucene syntax: “titulo:rex OR cargo:escrivao”. The Lucene’s scoring system —IndexSearcher— is then used to obtain the search results.

7.3 Structured queries

The second search modality takes advantage of the fact that the descriptive metadata, i.e the DUs, are highly structured. The specialized users, for which detailed contextual metadata is valuable, can benefit from this fact when searching for multimedia items. Such metadata can be easily indexed with the state-of-the-art indexes available in any relational database. MetaMedia has a dedicated interface for structured queries which is presented in Figure 7.4. The query values input in this interface are used to form traditional SQL queries. For example, a query made by an expert user, such as an archivist, wishing to find a DU dated on the “first of June 1250” that reached the digital archive via “Arquivo Distrital do Porto” would look like in the Figure 7.5.



Earliest Date : 1250/06/01

Source : Arquivo Distrital

Resultados: **2** UDs econtradas

1 PT/ADP/IED/DP/CSP/O/001A Documentos Title: - ShortTitle: Carta de confirmação de 1305-10-12

2 PT/IANTT/IED/DC/CSC/CAPC/012A Documentos Title: - ShortTitle: Resposta dada em Cortes ao bispo do Porto

Figure 7.4: The structured search interface

```
SELECT * FROM description_units WHERE
(earliestdate > to_date('1250/06/01','yyyy/mm/dd'))
and upper(source) LIKE upper('%Arquivo Distrital%')
```

Figure 7.5: An SQL query example

Given that some of the DUs data are also indexed with text indexes, as shown in Section 7.2, the relational indexes are redundant for the common set of data. However, having a set of basic data accessible with different query modalities it is beneficial for the retrieval efficiency. On one hand, it is a common pattern of use, even for non-specialized users, to first obtain some answer with keyword-based search and then to refine their needs by means of specific fields. On the

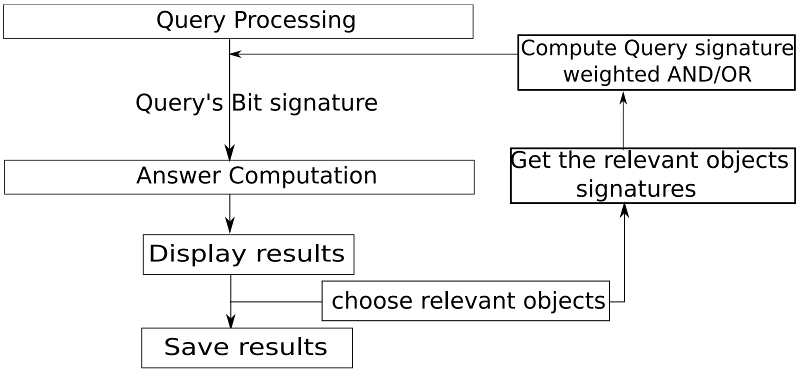


Figure 7.6: The Content-based Search Process

other hand, for some datatypes (from the common set of data) such as the numeric and date, the relational indexes such as B-trees, perform much better than text indexes. Therefore, indexing such datatypes with relational indexes is justified.

7.4 Query by content

The third search modality is query-by-content using audio-visual features. It relies on query-by-example and relevance feedback search paradigms. The search starts with a collection of query examples and the goal is to retrieve similar items. The similarity/dissimilarity evaluation can be computed with respect to some descriptor or to combinations of them. We have used a large set of low and high-level features. The low-level features include color, texture, shape and audio features, while the high-level features are concepts resulted from automatic annotation processes.

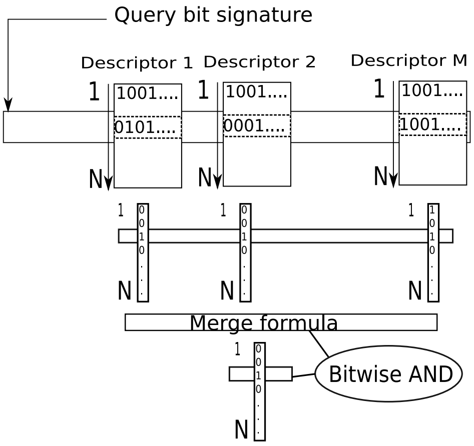


Figure 7.7: Answer computation

The color features are captured by several MPEG-7 descriptors, namely ColorLayout, ColorStructure, ScalableColor and ColorMoments; the texture is represented with EdgeHistogram, Homogeneous Texture, Wavelet texture and Haralick Texture descriptors, and the shape with the

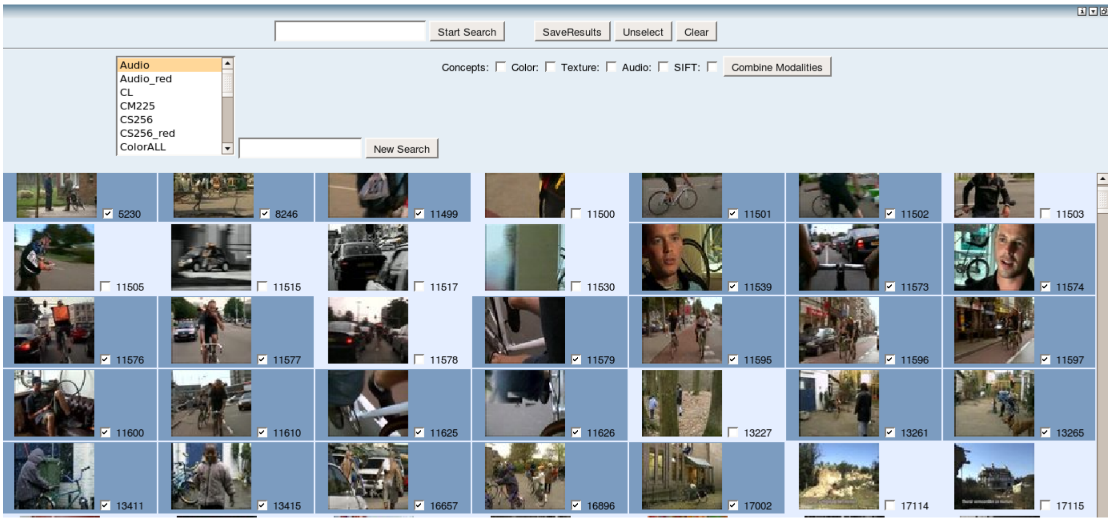
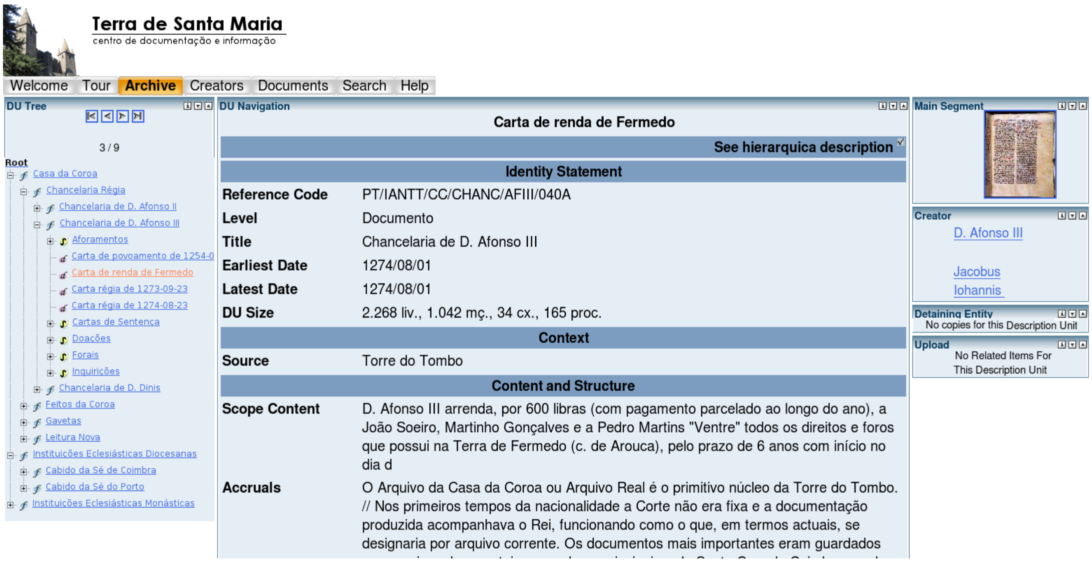


Figure 7.8: The audio-visual search Interface

RegionShape descriptor. We have also used keypoints, or local interest points, which are captured by the Scale Invariant Feature Transform (SIFT) descriptors [Low03]. Beside image features, we have used a set of automatically extracted descriptors for audio: Spectral Centroid, Spectral Rollof Point, Spectral Flux, Compactness, Spectral Variability, Root Mean Square, Fraction of Low Energy Windows, Zero Crossings, Strongest Beat, Beat Sum, Strength of Strongest Beat, MFCC (Mel Frequency Cepstral Coefficient), LPC (Linear Predictive Coding) and Area Method of Moments.



Carta de renda de Fermedo

See hierarquica description

Identity Statement

Reference Code	PT/IANTT/CC/CHANC/AFIII/040A
Level	Documento
Title	Chancelaria de D. Afonso III
Earliest Date	1274/08/01
Latest Date	1274/08/01
DU Size	2.268 liv., 1.042 mç., 34 cx., 165 proc.

Context

Source	Torre do Tombo

Content and Structure

Scope Content	D. Afonso III arrenda, por 600 libras (com pagamento parcelado ao longo do ano), a João Soeiro, Martinho Gonçalves e a Pedro Martins "Ventre" todos os direitos e foros que possui na Terra de Fermedo (c. de Arouca), pelo prazo de 6 anos com início no dia d
Accruals	O Arquivo da Casa da Coroa ou Arquivo Real é o primitivo núcleo da Torre do Tombo. // Nos primeiros tempos da nacionalidade a Corte não era fixa e a documentação produzida acompanhava o Rei, funcionando como o que, em termos actuais, se designaria por arquivo corrente. Os documentos mais importantes eram guardados

Creator: D. Afonso III; Jacobus Iohannis

Detaining Entity: No copies for this Description Unit

Upload: No Related Items For This Description Unit

Figure 7.9: Exploratory search

The high-level features, such as the concepts automatically obtained with special detectors

[JCLZ05, JNY07], are used to create a high-level descriptor, from now on referred to as the "Concepts" descriptor. Similarly to the work in [Rau04], we first create a concept space —a vector space where each concept is a dimension. Our concept space can be formalized as $C^n = \{c_1, c_2, .., c_n\}$, where c_i is a qualitative dimension corresponding to the i^{th} concept and n is the total number of concepts. Given this formalization, each of the annotated objects can be represented as vectors: $(v_1, v_2, .., v_n)$, where each v_i represents the value for the i^{th} corresponding conceptual dimension. The advantage of the "Concepts" descriptor is that it can be indexed it with multidimensional techniques. That allows us to use the BitMatrix — the multidimensional indexing technique that was presented in Chapter 6— for the entire range of low and high-level descriptors.

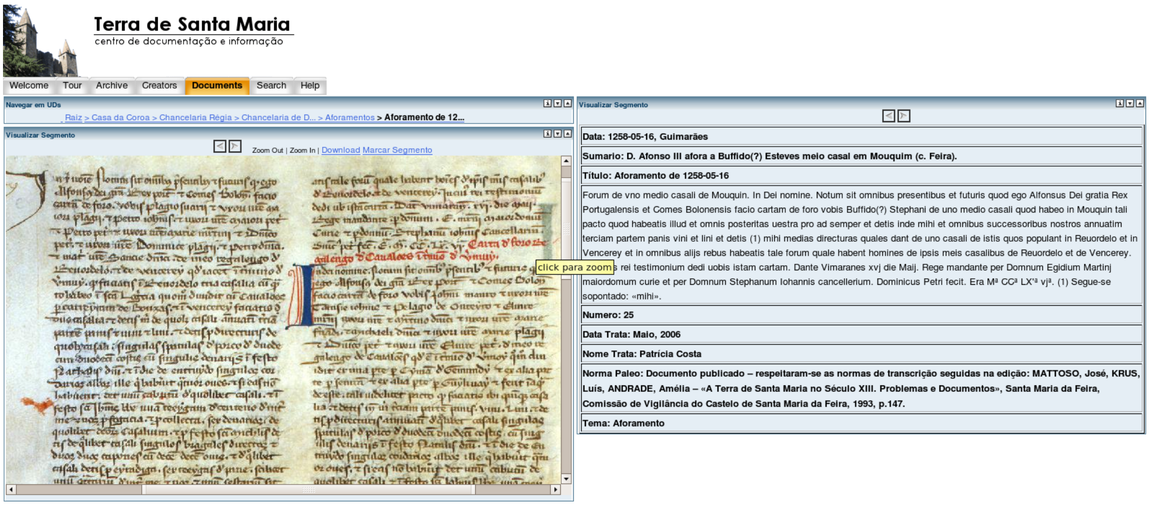
The content-based search starts with image or video examples and a set of descriptors to be used in the similarity computation. For example, it can start with an image of a natural scene and expect similar images with respect to the ColorLayout and ColorStructure (color), and EdgeHistogram (texture) descriptors. Figure 7.6 illustrates the overall process. The first step is to extract the desired descriptors from the example image, obtaining its feature vectors. In the sequel, the vectors are translated into bit signatures and passed to the BitMatrix-based answer computation step, which is illustrated in Figure 7.7.

Our answer computation strategy is to construct descriptor-wise BitMatrix indexes. At search time, depending on the selected descriptors, the corresponding BitMatrix indexes are searched, obtaining partial result lists which are then aggregated. The aggregation strategy is a configurable aspect of the content-based search. It can be either one of the aggregation strategies presented in Section 3.3.5, or a BitMatrix-based aggregation. The latter assumes that the partial result lists are binary, stating whether the objects are above a similarity threshold, or not. The descriptor-wise binary lists are aggregated with bitwise operations, as shown in Figure 7.7. As the case in the figure shows, all the lists have 1 for the object in the third position, thus the bitwise AND yields a 1 in the final list. Therefore, the third object will be among the retrieved objects.

The results are displayed on a wide scrollable surface as depicted in Figure 7.8 and can be marked for subsequent relevance feedback iterations. The interface also allows the customization of the descriptor set used in each search iteration. For example, the search can start with color and texture and then switch to audio descriptors.

7.5 Exploratory search

The last search modality is called exploratory search; its dedicated interface is illustrated in Figure 7.9. On the left-hand side, a tree-based control reflecting the hierarchical structure of the DUs can be observed. The user starts the DB exploration by browsing the tree and choosing a desired DU. With tree node expansions/collapses the user can further zoom in or out over the entire collection. Once a DU is selected, its sibling DUs can be sequentially accessed. This type of exploration is especially efficient when combined with the other search techniques presented up to now in this chapter.


Terra de Santa Maria
centro de documentação e informação
Welcome
Tour
Archive
Creators
Documents
Search
Help
Navegar em UDs
Raiz > Casa da Coroa > Chancelaria Régia > Chancelaria de D... > Aforamentos > Aforamento de 12...
Visualizar Segmento
Zoom Out | Zoom In | Download Marcar Segmento
click para zoom
Visualizar Segmento

Data: 1258-05-16, Guimarães

Sumario: D. Afonso III afora a Buffido(?) Esteves meio casal em Mouquim (c. Feira).

Título: Aforamento de 1258-05-16

Forum de vno medio casali de Mouquin. In Dei nomine. Notum sit omnibus presentibus et futuris quod ego Alfonsus Dei gratia Rex Portugalensis et Comes Bolonensis facio cartam de foro vobis Buffido(?) Stephani de uno medio casali quod habeo in Mouquin tali pacto quod habeatis illud et omnis posteritas uestra pro ad semper et detis inde mihi et omnibus successoribus nostros annuatim terciam partem panis vini et lini et detis (1) mihi medias directuras quales dant de uno casali de istis quos populant in Reuordelo et in Vencerey et in omnibus alijs rebus habeatis tale forum quale habent homines de ipsis meis casalibus de Reuordelo et de Vencerey. s rei testimonium dedi uobis istam cartam. Dante Vimaranes xvj die Maij. Rege mandante per Domnum Egidium Martinj maiordomum curie et per Domnum Stephanum Iohannis cancellerium. Dominicus Petri fecit. Era Mª CCª LX'ª vjª. (1) Segue-se sopontado: «mihi».

Numero: 25

Data Trata: Maio, 2006

Nome Trata: Patrícia Costa

Norma Paleo: Documento publicado – respeitaram-se as normas de transcrição seguidas na edição: MATTOSO, José, KRUS, Luís, ANDRADE, Amélia – «A Terra de Santa Maria no Século XIII. Problemas e Documentos», Santa Maria da Feira, Comissão de Vigilância do Castelo de Santa Maria da Feira, 1993, p.147.

Tema: Aforamento

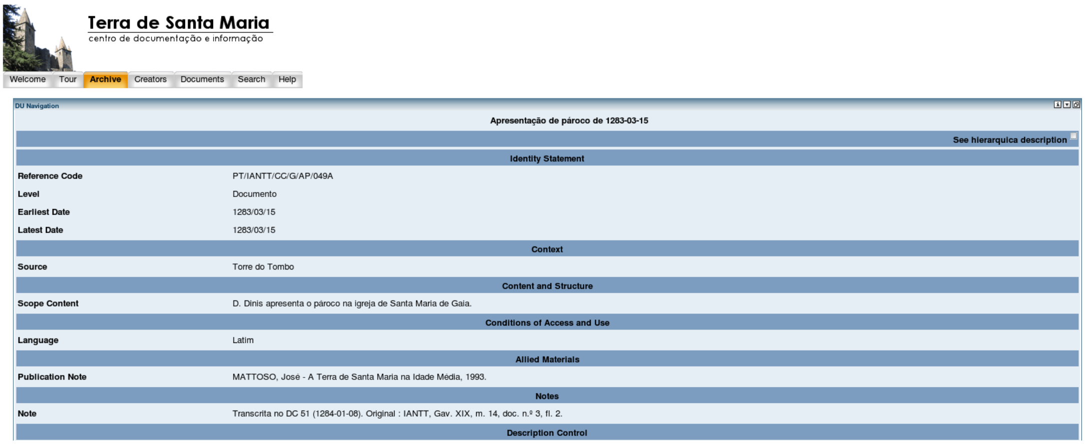
Figure 7.10: The content view

7.6 Faceted retrieval

We have seen several search modalities supported by MetaMedia. Each of them returns heterogeneous item parts. For example, some text results of a keyword search may represent an item's content, while others may be metadata. It is important to distinguish and display them in an informative manner. From the point of view of the relationship with the items, the search results can be content, context metadata and structure information. We regard these categories as facets of any multimedia item. Retrieving multimedia items from this content-description-structure perspective requires a previous identification, or categorization of data as being content, description or structure. This happens when the items are stored in the database, as a consequence of using the MetaMedia database model.

For each of the facets, MetaMedia offers dedicated views that are populated with data that comes from corresponding parts of the data model. Thus, the structural view displays hierarchies of DUs, the content view displays Segments and the description view displays the metadata stored in DUs. Note that the views follow the conceptual separation induced by the data model, which is orthogonal to the datatype-centric separation that is required for indexing. In this way, the search engines continue to index specific data types (structured data, text, image, video) but at retrieval time the results are properly displayed in the corresponding view using their location in the the data model. For example the text, image and video data that are stored in the database as Segments are all displayed in the content view, while the text stored in the DUs are displayed in the description view.

The faceted retrieval process starts with any of the available search modalities —keyword, structured, audio-visual or exploratory, and a first set of results is obtained. Based on the data model and independently of their media type, the results can be either content parts — Segments or descriptive metadata — DUs. The selection of a result identifies the multimedia item to which



Apresentação de pároco de 1283-03-15

Identity Statement	
Reference Code	PT/IANTT/CC/G/AP/049A
Level	Documento
Earliest Date	1283/03/15
Latest Date	1283/03/15
Context	
Source	Torre do Tombo
Content and Structure	
Scope Content	D. Dinis apresenta o páraco na igreja de Santa Maria de Gaia.
Conditions of Access and Use	
Language	Latim
Allied Materials	
Publication Note	MATTOSO, José - A Terra de Santa Maria na Idade Média, 1993.
Notes	
Note	Transcrita no DC 51 (1284-01-08). Original : IANTT, Gav. XIX, m. 14, doc. n.º 3, fl. 2.
Description Control	

Figure 7.11: The compact description view

it belongs and updates all the views of the MetaMedia interface. Thus, the selected item can be analyzed simultaneously in its several facets: content, description and structure. The item context is maintained across all the views. For example, after displaying the content in content view, the user can switch to structure view or descriptive view and observe the hierarchy of the same item.

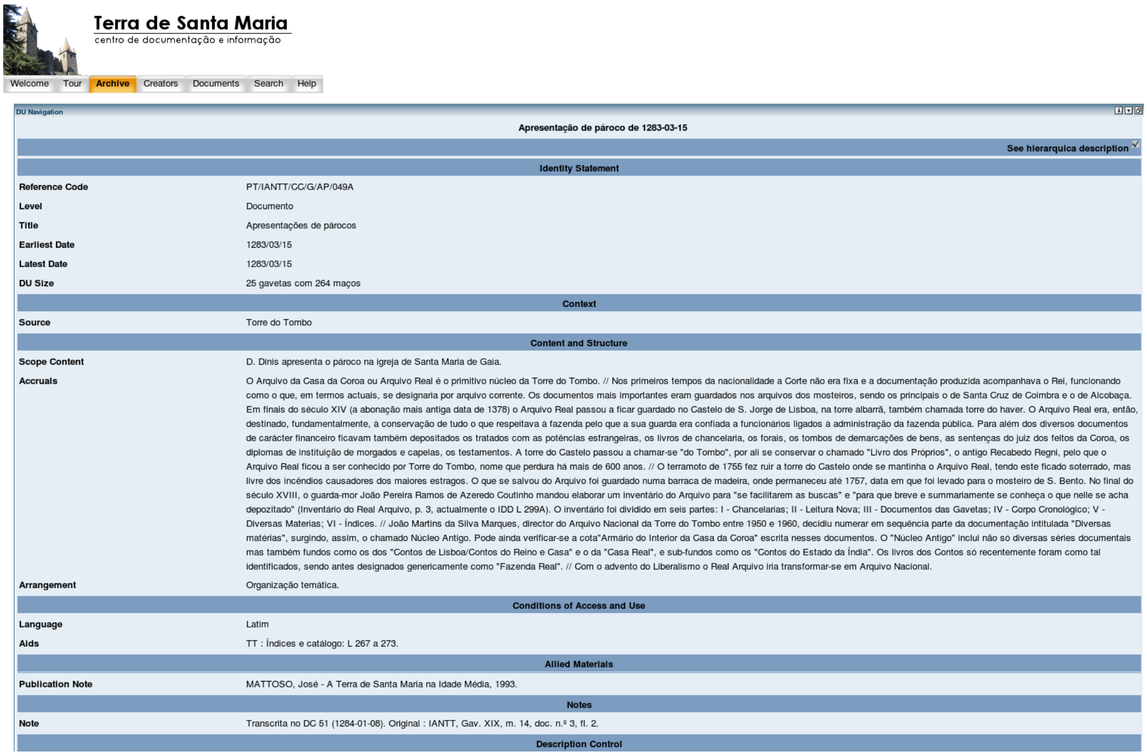
Figure 7.10 illustrates the content view interface for a collection of historic documents (detailed in Chapter 8). On the left-hand side, there is a viewer for image segments and on the right-hand side the text segments are displayed. Both visualization areas can be maximized to allow the focus on the selected piece of content.

Figures 7.11 and 7.12 illustrate the description view. This view has a remarkable feature. Given the uniform description of the DUs hierarchy, the same descriptive attributes are found at all the levels. The user can visualize the attributes of a selected DU in two modes. The first one presents only the non-empty attributes at the DU level, while the second mode illustrates the full set of attributes, filling the unspecified ones with values inherited from upper levels. In Figures 7.11 and 7.12 capture both the compact and full visualization modes.

The structural view is a panel where the hierarchical organization of the items can be visualized. Illustrated on the left part of Figure 7.9, this view is especially useful for the visualization of complex items, i.e items that contain sub-items. The user can navigate down to sub-items (*zoom*) or up to the root obtaining an overview of the entire archive.

7.7 Discussion

We have presented in this chapter the retrieval approach that MetaMedia offers. There are search methods for text, structured, image, video and high-dimensional data, each one having its own search interface. Besides specialized search interfaces, there are three facets, also called views, which have been designed to display data from specific parts of the data model. The content view displays Segments, which are also considered the multimedia items content. The description view displays the items contextual metadata (Description Units) and the structure view reflects



Apresentação de pároco de 1283-03-15

See hierarquica description ✓

Identity Statement	
Reference Code	PT/IANTT/CC/G/AP/049A
Level	Documento
Title	Apresentações de párocos
Earliest Date	1283/03/15
Latest Date	1283/03/15
DU Size	25 gavetas com 264 maços
Context	
Source	Torre do Tombo
Content and Structure	
Scope Content	D. Dinis apresenta o pároco na igreja de Santa Maria de Gaia.
Accruals	O Arquivo da Casa da Coroa ou Arquivo Real é o primitivo núcleo da Torre do Tombo. // Nos primeiros tempos da nacionalidade a Corte não era fixa e a documentação produzida acompanhava o Rei, funcionando como o que, em termos actuais, se designaria por arquivo corrente. Os documentos mais importantes eram guardados nos arquivos dos mosteiros, sendo os principais o de Santa Cruz de Coimbra e o de Alcobaça. Em finais do século XIV (a abonação mais antiga data de 1378) o Arquivo Real passou a ficar guardado no Castelo de S. Jorge de Lisboa, na torre albarrã, também chamada torre do haver. O Arquivo Real era, então, destinado, fundamentalmente, à conservação de tudo o que respeitava à fazenda pelo que a sua guarda era confiada a funcionários ligados à administração da fazenda pública. Para além dos diversos documentos de carácter financeiro ficavam também depositados os tratados com as potências estrangeiras, os livros de chancelaria, os forais, os tombos de demarcações de bens, as sentenças do juiz dos feitos da Coroa, os diplomas de instituição de morgados e capelas, os testamentos. A torre do Castelo passou a chamar-se "do Tombo", por ali se conservar o chamado "Livro dos Próprios", o antigo Recabedo Regni, pelo que o Arquivo Real ficou a ser conhecido por Torre do Tombo, nome que perdura há mais de 600 anos. // O terramoto de 1755 fez ruir a torre do Castelo onde se mantinha o Arquivo Real, tendo este ficado soterrado, mas livre dos incêndios causadores dos maiores estragos. O que se salvou do Arquivo foi guardado numa barraca de madeira, onde permaneceu até 1757, data em que foi levado para o mosteiro de S. Bento. No final do século XVIII, o guarda-mor João Pereira Ramos de Azeredo Coutinho mandou elaborar um inventário do Arquivo para "se facilitarem as buscas" e "para que breve e summariamente se conheça o que nelle se acha depozitado" (Inventário do Real Arquivo, p. 3, actualmente o IDD L 299A). O inventário foi dividido em seis partes: I - Chancelarias; II - Leitura Nova; III - Documentos das Gavetas; IV - Corpo Cronológico; V - Diversas Materias; VI - Índices. // João Martins da Silva Marques, director do Arquivo Nacional da Torre do Tombo entre 1950 e 1960, decidiu numerar em sequência parte da documentação intitulada "Diversas matérias", surgindo, assim, o chamado Núcleo Antigo. Pode ainda verificar-se a cota"Armário do Interior da Casa da Coroa" escrita nesses documentos. O "Núcleo Antigo" inclui não só diversas séries documentais mas também fundos como os dos "Contos de Lisboa/Contos do Reino e Casa" e o da "Casa Real", e sub-fundos como os "Contos do Estado da Índia". Os livros dos Contos só recentemente foram como tal identificados, sendo antes designados genericamente como "Fazenda Real". // Com o advento do Liberalismo o Real Arquivo iria transformar-se em Arquivo Nacional.
Arrangement	Organização temática.
Conditions of Access and Use	
Language	Latim
Aids	TT : Índices e catálogo: L 267 a 273.
Allied Materials	
Publication Note	MATTOSO, José - A Terra de Santa Maria na Idade Média, 1993.
Notes	
Note	Transcrita no DC 51 (1284-01-08). Original : IANTT, Gav. XIX, m. 14, doc. n.º 3, fl. 2.
Description Control	

Figure 7.12: The full description view

the hierarchical structure of the items as it is introduced by the relationship between Description Units.

The set of interfaces allow the user to combine operations such as *overview* and *zoom* by interacting with the tree visualization structure and perform *filter* operations by resorting to the several search types. Once entered in a retrieval session, the retrieval experience is user dependent. For a novice user for example, perhaps overview has more importance than to an expert. The latter could quickly use filter and zoom to find the desired documents.

Chapter 8

Case Studies

This chapter presents two case studies, “Enthrone” and “Santa Maria da Feira”. Both case studies make intensive use of metadata and are implemented on top of the same MetaMedia data model. However, they provide different perspectives on the datasets bringing challenges from the point of view of structural organization, visualization and search.

Metadata standards have been proposed at a fast pace [SS06], but their adoption in practical systems has not followed the same trend. Some argued that it is time to stop developing metadata standards and build systems based on the existing models even if they are not perfect [Bul04]. We need examples of systematic use of the metadata standards, where the specificity of the application domains gives insight on the appropriateness of the concepts used. The case studies presented here are two examples of such systems.

8.1 Structure of the case studies

The two case studies are quite different in nature. The “Enthrone” case study is focused on quality of service for news broadcasting and video on demand scenarios in heterogeneous networks, while “Santa Maria da Feira” aims at an integrated view of digitized heritage documents, archival descriptions and transcription texts produced by scholars. The following two sections cover the “Enthrone” and “Santa Maria da Feira” case studies, respectively. Each of them is structured in three subsections that present the nature of the multimedia items, how are they mapped to our data model and the retrieval methods. The items are regarded as instances of the generic multimedia items, previously presented in Section 2.3.3. Then, based on items analysis, we identify the amount of data that goes into the relational and XML parts of the model, and the proper indexing strategies. Finally, the “Retrieval” sections specify which of the retrieval methods presented in Chapter 7 have been applied in each case.

8.2 Enthrone, news broadcasting and video on demand

The MetaMedia content-management solution has been adopted in news broadcasting and video on demand scenarios in the context of the ENTHRONE (End-to-End QoS through Integrated Management of Content, Networks and Terminals) project [The, CAA+07].

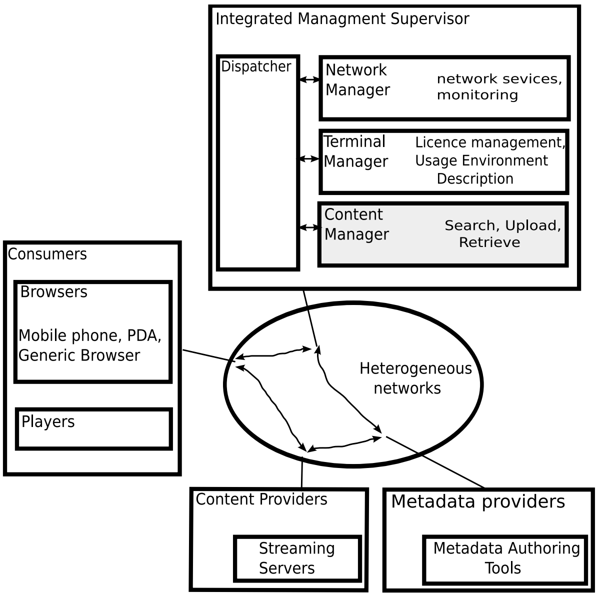


Figure 8.1: Enthrone

The goal of the project is to provide integrated management for the access to multimedia content with end-to-end quality of service (QoS) in heterogeneous networks and terminals. That is to make available any multimedia content to the ever increasing number of different users with different preferences who access it through a plethora of devices and over heterogeneous networks. End devices range from mobile phones to high definition TVs, access networks can be as diverse as GSM and broadband networks, and the various backbone networks are different in bandwidth and QoS support. Moreover, the users have different content/presentation preferences and intend to consume the content at different locations, times, and under varying circumstances.

Figure 8.1 presents a high-level view of the actors involved in Enthrone: content providers, metadata providers, Integrated Management Supervisor (IMS) and consumers. IMS is the core system that takes care of the integration, comprising several sub-systems dedicated to the management of the network, the terminals (consumers) and the content. The IMS needs to be aware of the entire audio-visual service distribution chain including content generation and protection, search, adaptation, distribution across networks and reception at user terminals. IMS includes a Content Manager sub-system used to efficiently store, search, retrieve and adapt the set of metadata descriptions that accompany the whole distribution chain.

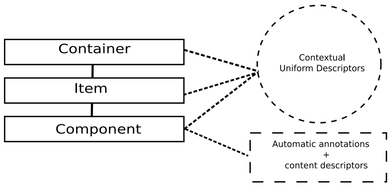


Figure 8.2: The Levels of the Hierarchy

8.2.1 Items

Broadcasting and video on-demand are the main services that ENTHRONE offers. The consumable items range from simple audiovisual fragments to structured collections thereof. For example, an item can be a piece of news about traffic conditions, including its text, audio and video tracks, or a video obtained from a video-on-demand provider. Collections of items, such as a news program from a broadcaster, are also perceived as items. The Enthrone's processing chain must hand a large variety of metadata. Accompanying collections of raw content with descriptors that cover identification, search, digital rights, user-terminals and adaptation, produces complex multimedia items that we call *Enthrone digital items*.

The Enthrone digital items are well-formed XML files, with the overall structure imposed by the MPEG-21 DID Schema [MPE04]. The DID Schema, is an MPEG-21 XML Schema which defines the Digital Item Declaration Language (DIDL), i.e. the set of rules necessary for the definition of the MPEG-21 Digital Items. Below we present some of the elements:

- *DIDL*: is the root element of a DID.
- *Container*: groups more Items and/or sub-Containers.
- *Item*: contains Components and/or sub-Items. Item is seen in MPEG-21 as the lowest level of granularity transacted by users.
- *Component*: binds resources to a set of descriptions.
- *Descriptor*: associates information with the enclosing element (*Item*, *Container*, *Component*, *Descriptor*).

The Enthrone digital items follow in fact a profile of the MPEG-21 DID Schema. We call it a profile because, we have introduced additional constraints to the generic MPEG-21 DID Schema by defining the number of child elements allowed at specific description levels and the identification rules. For the example the root element *DIDL* must contain one and only one direct child which must be of type *Container*. An MPEG-21 Digital Item Identification descriptor containing an unique URI is mandatory at the level of Container. The *Item* and *Component* types have similar restrictions. *Descriptor* can be applied to any *Container*, *Item* or *Component*, but not to an

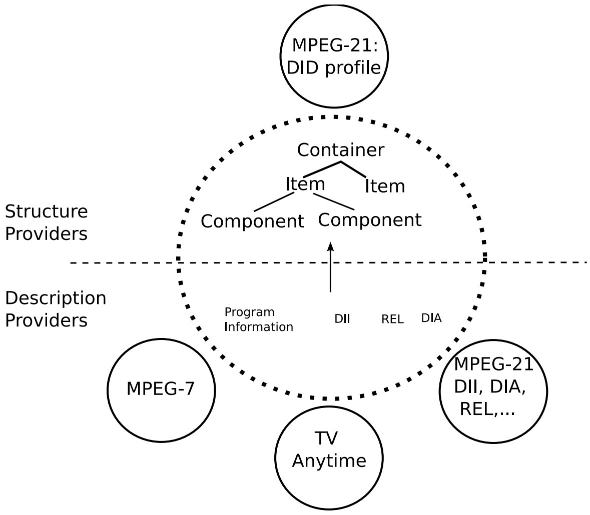


Figure 8.3: The Enthrone digital items

enclosing *Descriptor*. The structure of the Enthrone digital items is illustrated in Figures 8.2 and 8.3.

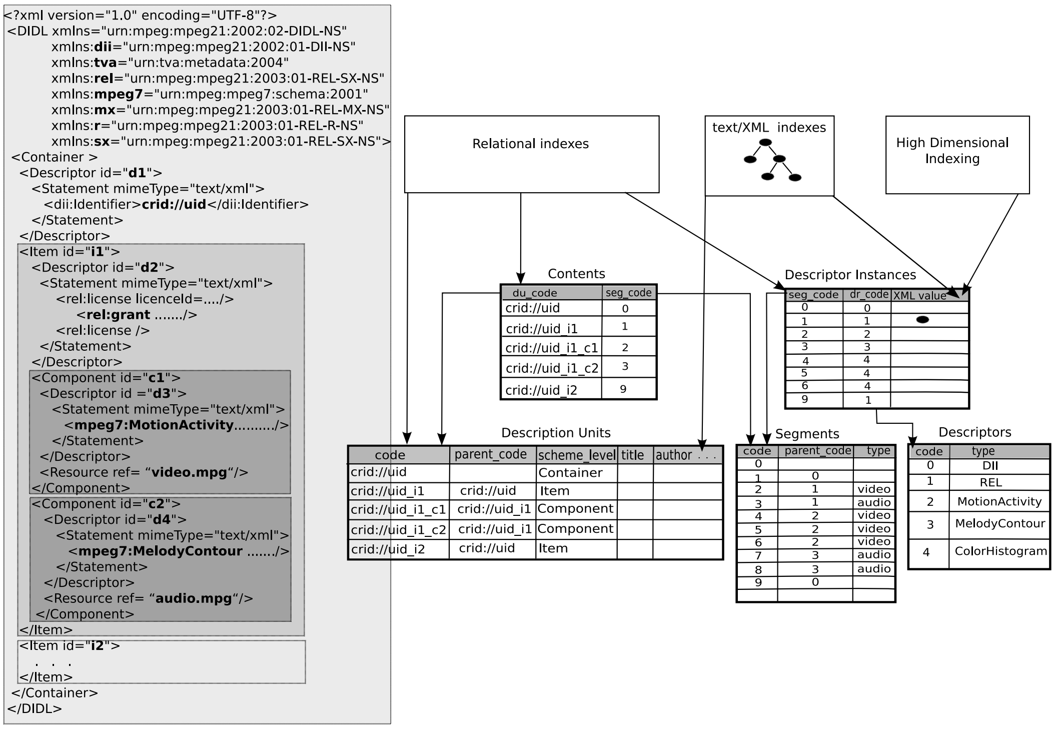
Figure 8.3 illustrates the Enthrone digital items by particularizing the generic structure that has been illustrated in Figure 2.2.a. It can be observed that MPEG-21 DID plays the role of the structure provider for the Enthrone dataset, while TV-Anytime, MPEG-21 and MPEG-7 are description providers. The most used descriptors with provenance in the description providers are: “ProgramInformation” and “Schedule” descriptors from TV-Anytime, “Digital Item Identification”, “Digital Item Adaptation” and “Rights Expression Language” descriptors from MPEG-21 and “ColorHistogram”, “MelodyContour” and “MotionActivity” from MPEG-7.

8.2.2 Model

The data model proposed in Chapter 5 has been instantiated in the Enthrone case study [CAA+07]. We present in this section how the previously presented Enthrone digital items have been mapped to the database model. Given the concepts behind the data model, Figure 8.4 shows how such items are captured in DUs, Segments and Descriptors. The DUs capture the structural organization which is given by the items structure provider (Figures 8.2 and 8.3). The left part of Figure 8.4 illustrates an Enthrone digital item in XML format as it appears before being stored in the database, while the right part of the same Figure shows what data is explicitly stored in the database.

For each of the *Container*, *Item* or *Component* elements there will be a corresponding DU, i.e a corresponding entry in the “Description Units” table. The part-of relations between DUs are established by the `code` and `parent_code` attributes, which have their values assigned from the *id* attributes of each element.

The content resources of the Enthrone digital items appear below the *Component* elements. They are mapped to Segments in the data model and in general are not independently consumed; their identifiers are stored in the “Segments” table. The Segments contextual metadata is taken



```
<?xml version="1.0" encoding="UTF-8"?>
 <DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
          xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
          xmlns:tva="urn:tva:metadata:2004"
          xmlns:rel="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
          xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
          xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
          xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
          xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS">
  <Container >
   <Descriptor id="d1">
     <Statement mimeType="text/xml">
       <dii:Identifier>crid://uid</dii:Identifier>
     </Statement>
   </Descriptor>
   <Item id="i1">
     <Descriptor id="d2">
       <Statement mimeType="text/xml">
          <rel:license licenceId=..../>
              <rel:grant ......./>
          <rel:license />
       </Statement>
     </Descriptor>
     <Component id="c1">
       <Descriptor id ="d3">
         <Statement mimeType="text/xml">
           <mpeg7:MotionActivity........../>
         </Statement>
       </Descriptor>
       <Resource ref= “video.mpg“/>
     </Component>
     <Component id="c2">
        <Descriptor id="d4">
          <Statement mimeType="text/xml">
            <mpeg7:MelodyContour ......./>
          </Statement>
        </Descriptor>
        <Resource ref= “audio.mpg“/>
      </Component>
   </Item>
   <Item id="i2">
      .   .   .
   </Item>
  </Container>
 </DIDL>
```

Contents

du_code	seg_code
crid://uid	0
crid://uid_i1	1
crid://uid_i1_c1	2
crid://uid_i1_c2	3
crid://uid_i2	9

Descriptor Instances

seg_code	dr_code	XML value
0	0	
1	1	●
2	2	
3	3	
4	4	
5	4	
6	4	
9	1	

Description Units

code	parent_code	scheme_level	title	author . . .
crid://uid		Container		
crid://uid_i1	crid://uid	Item		
crid://uid_i1_c1	crid://uid_i1	Component		
crid://uid_i1_c2	crid://uid_i1	Component		
crid://uid_i2	crid://uid	Item		

Segments

code	parent_code	type
0		
1	0	
2	1	video
3	1	audio
4	2	video
5	2	video
6	2	video
7	3	audio
8	3	audio
9	0	

Descriptors

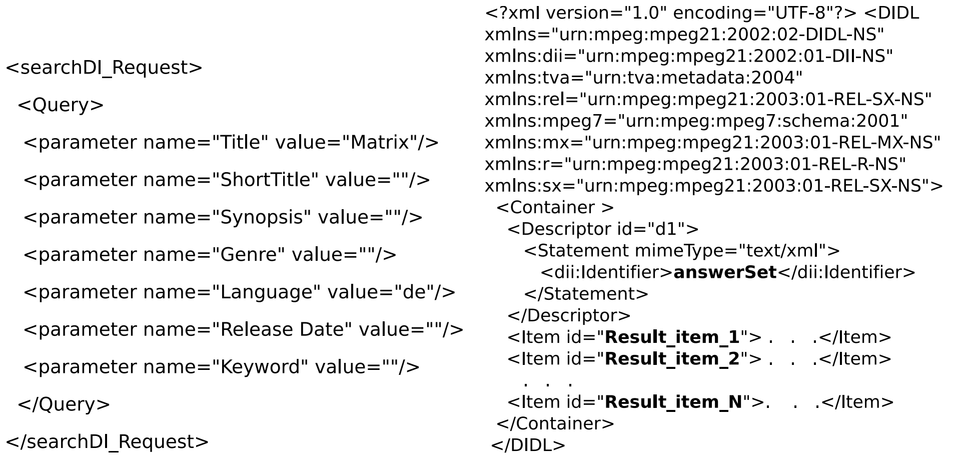
code	type
0	DII
1	REL
2	MotionActivity
3	MelodyContour
4	ColorHistogram

Figure 8.4: Storing Enthrone digital items

from their corresponding DUs. The association between a Segment and a DU, is made explicit with the “Contents” table. That is, the “Contents” table must contain a record with two codes: the segment code (`seg_code`) and the DU code (`du_code`). The Segments can also be hierarchically organized. This happens when the content resources require further analysis at even more fine-grained levels. For example, a video resource from a digital item which is a Segment in our data model, is selected for analysis at frame level. The frames become sub-segments and inherit their context metadata from the DU of their parent Segment. The Segments hierarchies are also modeled with `code` and `parent_code` attributes.

The Descriptors part of the model hosts information provided by the standards identified as description providers (see Figure 8.3). In Figure 8.4, the “Descriptors” table contains references to all the descriptor types that appear in the example digital item. The “Descriptor_Instances” table accounts for the associations between Segments and Descriptors, similarly to the role played by the “Contents” table for DUs and Segments.

The identifiers are required to differentiate elements of the same type within an Enthrone digital item and to uniquely identify them in the database. Generally, the identifiers are derived from two mandatory identification elements: the MPEG-21 DII descriptor and the *id* attributes. The `code` column of the “Description Units” table in Figure 8.4 shows the identifiers for the example Enthrone digital item. For example the identification code “crid:uid_i1” that corresponds to the first Item is formed by the concatenation of the top level DII value “crid:uid” and the *id* attribute “i1”.



```
<searchDI_Request>
 <Query>
  <parameter name="Title" value="Matrix"/>
  <parameter name="ShortTitle" value=""/>
  <parameter name="Synopsis" value=""/>
  <parameter name="Genre" value=""/>
  <parameter name="Language" value="de"/>
  <parameter name="Release Date" value=""/>
  <parameter name="Keyword" value=""/>
 </Query>
</searchDI_Request>
```

(a) A search request in XML format

```
<?xml version="1.0" encoding="UTF-8"?> <DIDL
xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
xmlns:tva="urn:tva:metadata:2004"
xmlns:rel="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS">
 <Container >
  <Descriptor id="d1">
    <Statement mimeType="text/xml">
      <dii:Identifier>answerSet</dii:Identifier>
    </Statement>
  </Descriptor>
  <Item id="Result_item_1"> .  .  .</Item>
  <Item id="Result_item_2"> .  .  .</Item>
    .  .  .
  <Item id="Result_item_N">.   .  .</Item>
 </Container>
</DIDL>
```

(b) A result set in the MPEG-21 DID format

Figure 8.5: Query and Results in Enthrone

Figure 8.4 shows that the data model combines two storage types: relational and XML. The hierarchical structure of the example digital item, together with the contextual metadata are stored in the relational part of the database: “Description Units”, “Segments” and “Contents” tables. The descriptor types also go in the relational part, namely in the “Descriptors” table. But the instances of the descriptors are stored in XML. The “Descriptor_Instances” table with its column of “XMLType” is the place in the model that accommodates the XML descriptors, as illustrated in Figure 8.4.

8.2.3 Retrieval

The search and retrieval tasks in Enthrone are initiated through API calls by other sub-systems such as Consumers, Content Providers or Metadata Providers (see Figure 8.1). There have been no specifications for a dedicated retrieval interface.

Figure 8.4 shows that each data type, namely relational, text/XML, and high-dimensional, are handled by proper indexing methods. However, given that the queries are mostly focused on contextual metadata, which is located in the “Description Units” table, the relational and text indexes are the most important.

The queries are keyword-based and are sent to the query engine in XML format. There is a fixed list of search keywords taken from the TV-Anytime “ProgramInformation” descriptor. It contains the “title”, “genre”, “synopsis”, “date” and “summary” keywords.

An example of a query requesting all the items having title “Matrix” and language “German” is presented in Figure 8.5(a). After parsing the query XML, the sequence of keywords is obtained and a query is effectively issued to the query engine. The initial phase of the answer computation is to obtain the set DUs identifiers that match the given query. Next, the metadata located in these DUs is returned in MPEG-21 compliant format — a “virtual” Digital Item obtained from the

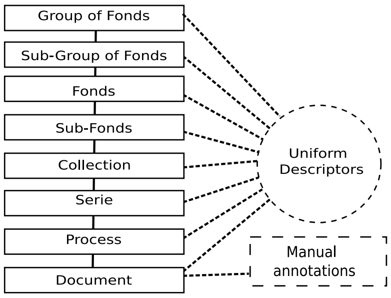


Figure 8.6: The Levels of the Hierarchy

results. Figure 8.5(b) details the answer format. The important MPEG-21 DID elements are *Item*, which appear for each of the matching DUs and the *Container*, in which all the *Item*s are bundled.

8.3 Santa Maria da Feira, a historic documentation center

The Santa Maria da Feira case study [RDC07a, RDC07b] has archivistic requirements, thus with great concern for authenticity, organization and preservation of documents. The original documents are parchments and the documentation center is a virtual archive based on digitized versions and their associated transcriptions.

We outline the process of producing the dataset which has been quite complex, requiring the participation of experts from the History, Archival, Linguistics and Information Retrieval domains. A first work thread was the digitization of the original parchments, which produced high-definition electronic versions thereof. In parallel, the historians had the task of producing the parchment transcriptions in either Latin or archaic Portuguese, and to annotate them. A dedicated annotation tool [RDB06] has been developed and a thesaurus has been defined in order to assist the transcriptions annotation process. Enriched transcriptions with labels such as “person”, “place”, “institution” or “date” have been thus obtained.

The archivists have focused on the organization of the dataset. Built according to the ISAD/-ISAAR standards, the dataset structure is based on the provenance of the individual documents. For example, documents resulting from the activity of a specific office, such as the king’s chancellor’s office, are kept together and grouped in description units. Beside structural organization, the description units contain also important amounts of contextual metadata, such as title, short title, date of creation, size, scope, accruals, arrangement and materials. However, not all the description units have detailed archival description. According to the archivists, there can be objects for which context metadata generation can be afforded, due to their importance, while many others will not justify such an investment. In any case, some minimal description is required. This can be achieved through hierarchical organization of the documents, where the descriptive information present at the collection level can be assumed for the individual items as well, with little or no effort. The hierarchical organization of the documents allows items to inherit descriptions created at collection level, in case they do not have specific ones already defined.

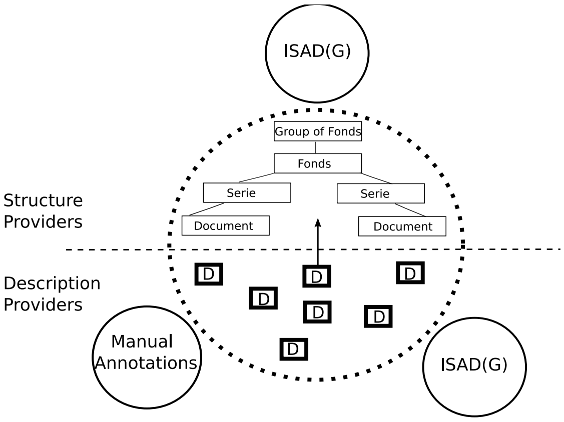


Figure 8.7: The Santa Maria da Feira items

Summarizing, after all the experts contributions, the information for each of the documents in the repository comprised four parts: the digitized parchment, its transcription, its annotated transcription and the archival description according to the ISAD/ISAAR standards. Towards making the digitizations and transcriptions publicly available, the development of a dedicated retrieval system has been identified as a mandatory task.

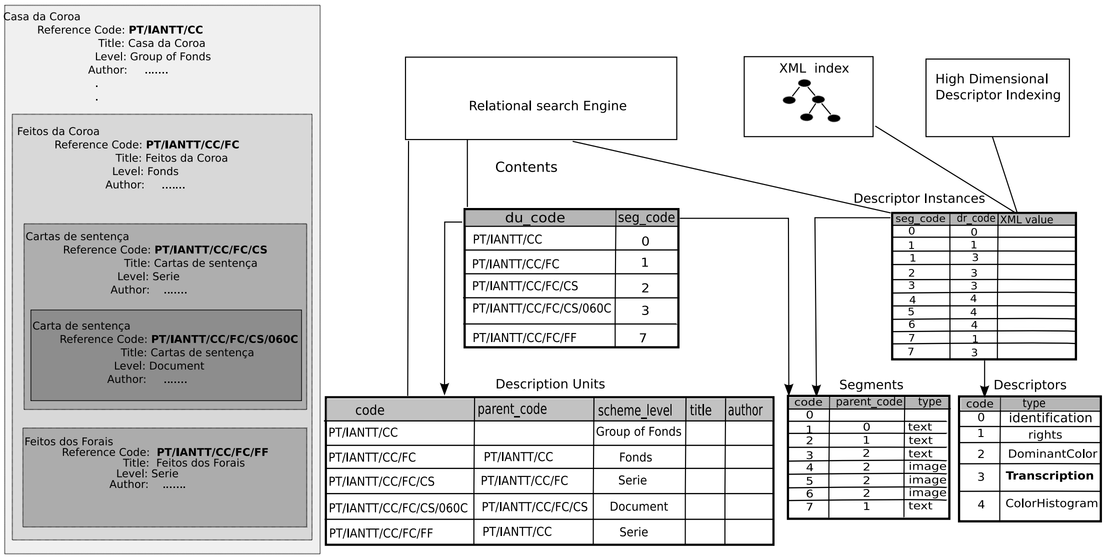
Items

All the electronic information available for a document which, as stated before, comprises the digitization, transcription, annotated transcription and the archival description, is regarded as a Santa Maria da Feira digital item. However, given the part-of hierarchical organization, sets of items can be grouped at higher descriptive levels. The results of such groupings are also seen as items.

Unlike the Enthrone case study, in Santa Maria da Feira the digital items do not come prepackaged in single files. The various information parts belonging to a digital item are introduced independently in the database. However, after having the items in the database, they could be exported in a specific format such as XML, joining all the metadata in one file.

The structure provider for this set of items is ISAD, the norm on which the archive has been organized. Figure 8.6 show that the items structure is composed of the *Group of Fonds*, *Sub-Group of Fonds*, *Fonds, Sub-Fonds*, *Collection*, *Series*, *Process* and *Document* levels.

The descriptive metadata come from two description providers. First, a set of attributes from the ISAD(G) norm has been used to convey contextual metadata. Attributes such as “code”, “title”, “author”, “date”, “copyright”, “material”, “scope”, “language”, “notes” are uniformly present at each of the hierarchical levels presented before. The second category of descriptive information, content-related in this case, has been provided by the manually annotated transcriptions. We remind here that the annotations have been obtained from the clean transcriptions which have been enriched in the sequel by archivists with concepts such as “person”, “noble title”, “date” or “institution”. The enriched transcriptions, are XML documents and have been incorporated as



Casa da Coroa
Reference Code: **PT/IANTT/CC**
Title: Casa da Coroa
Level: Group of Fonds
Author: ....
.
.

Feitos da Coroa
Reference Code: **PT/IANTT/CC/FC**
Title: Feitos da Coroa
Level: Fonds
Author: ....

Cartas de sentença
Reference Code: **PT/IANTT/CC/FC/CS**
Title: Cartas de sentença
Level: Serie
Author: ....

Carta de sentença
Reference Code: **PT/IANTT/CC/FC/CS/060C**
Title: Cartas de sentença
Level: Document
Author: ....

Feitos dos Forais
Reference Code: **PT/IANTT/CC/FC/FF**
Title: Feitos dos Forais
Level: Serie
Author: ....

du_code	seg_code
PT/IANTT/CC	0
PT/IANTT/CC/FC	1
PT/IANTT/CC/FC/CS	2
PT/IANTT/CC/FC/CS/060C	3
PT/IANTT/CC/FC/FF	7

seg_code	dr_code	XML value
0	0	
1	1	
1	3	
2	3	
3	3	
4	4	
5	4	
6	4	
7	1	
7	3	

code	parent_code	scheme_level	title	author
PT/IANTT/CC		Group of Fonds		
PT/IANTT/CC/FC	PT/IANTT/CC	Fonds		
PT/IANTT/CC/FC/CS	PT/IANTT/CC/FC	Serie		
PT/IANTT/CC/FC/CS/060C	PT/IANTT/CC/FC/CS	Document		
PT/IANTT/CC/FC/FF	PT/IANTT/CC	Serie		

code	parent_code	type
0		
1	0	text
2	1	text
3	2	text
4	2	image
5	2	image
6	2	image
7	1	text

code	type
0	identification
1	rights
2	DominantColor
3	**Transcription**
4	ColorHistogram

Figure 8.8: Storing Santa Maria da Feira digital items

descriptors; we call them "Transcription" descriptors. The XML Schema that such descriptors comply has been locally defined, thus non-standard. This is particularly important because it marks the use of a user-defined description provider.

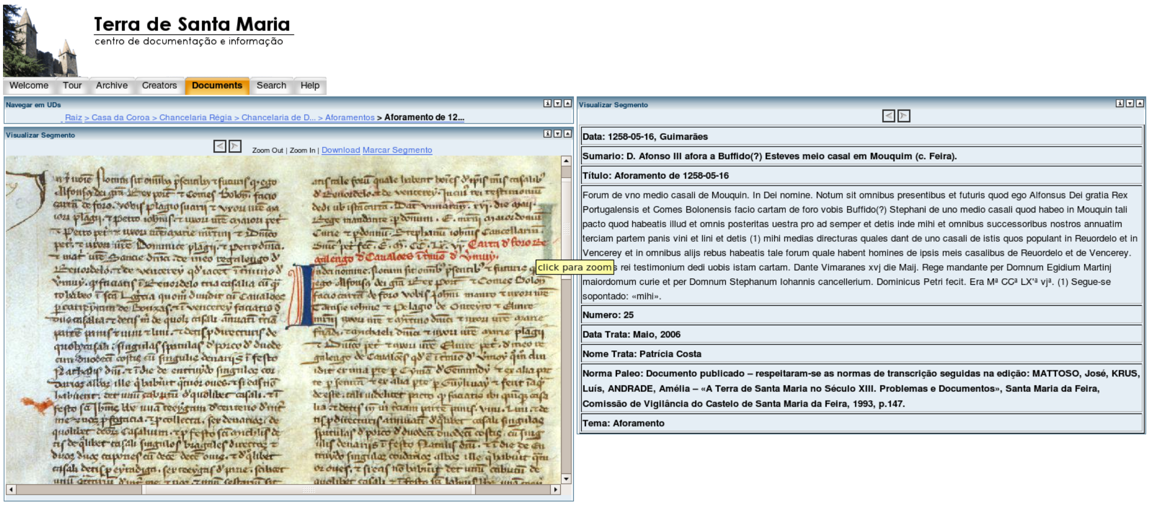
Figure 8.7 illustrates the structural and descriptive parts of the Santa Maria da Feira's multimedia items. The Figure follows the generic items structure that has been illustrated in Figure 2.2.a. It can be observes that ISAD(G) appears both as a structure and as a description provider. Also as description providers there are the enriched transcriptions.

Model

Each level of the hierarchy presented in Figure 8.6 has a corresponding DU. The "Description Units" table in Figure 8.8 contains separate instances for each of the DUs. Attributes such as "code", "title", "author", "date", "copyright", "material", "scope", "language", "notes" that uniformly apply at all the description levels, are also stored in the "Description Units" table. The DUs identifiers are being derived in this case from the reference codes established by archivists for the original parchments

The content-only parts of item are conceptualized as Segments and are stored in the table with the same name. Their association to the DUs is implemented with the help of the "Contents" table, by joining in each of the records a Segment and its corresponding DU code. Valid Segments examples are the digitizations and the transcriptions.

The enriched transcriptions, are incorporated in the model as instances of the "Transcription" descriptor. The association of Segments to Descriptors is made through the "Descriptor Instances" table as illustrated in Figure 8.8.


Terra de Santa Maria
centro de documentação e informação
Welcome
Tour
Archive
Creators
Documents
Search
Help
Navegar em UDs
Raiz > Casa da Coroa > Chancelaria Régia > Chancelaria de D... > Aforamentos > Aforamento de 12...
Visualizar Segmento
Zoom Out | Zoom In | Download Marcar Segmento
click para zoom
Visualizar Segmento

Data: 1258-05-16, Guimarães

Sumario: D. Afonso III afora a Buffido(?) Esteves meio casal em Mouquim (c. Feira).

Título: Aforamento de 1258-05-16

Forum de vno medio casali de Mouquin. In Dei nomine. Notum sit omnibus presentibus et futuris quod ego Alfonsus Dei gratia Rex Portugalensis et Comes Bolonensis facio cartam de foro vobis Buffido(?) Stephani de uno medio casali quod habeo in Mouquin tali pacto quod habeatis illud et omnis posteritas uestra pro ad semper et detis inde mihi et omnibus successoribus nostros annuatim terciam partem panis vini et lini et detis (1) mihi medias directuras quales dant de uno casali de istis quos populant in Reuordelo et in Vencerey et in omnibus alijs rebus habeatis tale forum quale habent homines de ipsis meis casalibus de Reuordelo et de Vencerey. s rei testimonium dedi uobis istam cartam. Dante Vimaranes xvj die Maij. Rege mandante per Domnum Egidium Martinj maiordomum curie et per Domnum Stephanum Iohannis cancellerium. Dominicus Petri fecit. Era Mª CCª LX'ª vjª. (1) Segue-se sopontado: «mihi».

Numero: 25

Data Trata: Maio, 2006

Nome Trata: Patrícia Costa

Norma Paleo: Documento publicado – respeitaram-se as normas de transcrição seguidas na edição: MATTOSO, José, KRUS, Luís, ANDRADE, Amélia – «A Terra de Santa Maria no Século XIII. Problemas e Documentos», Santa Maria da Feira, Comissão de Vigilância do Castelo de Santa Maria da Feira, 1993, p.147.

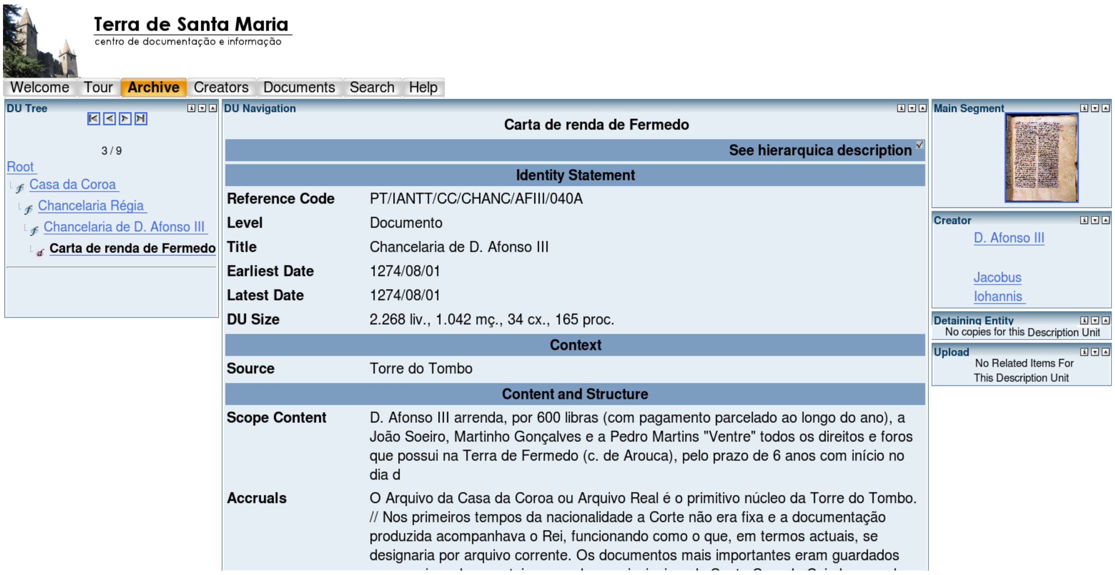
Tema: Aforamento

Figure 8.9: The content view

Retrieval

Search and retrieval can be performed by both specialized and lay users with the help of the MetaMedia portal interface. The various parts of Santa Maria da Feira's items, namely the digitizations, the transcriptions, the archivistic descriptions and the structure become retrievable facets of the items and have dedicated search methods.

The retrieval process relies on previous indexing steps that were taken type-wise. Figure 8.8 shows that each data type, namely relational data, text/XML and high-dimensional, are handled by proper indexing methods [CRD06].



Carta de renda de Fermedo

See hierarquica description

No copies for this Description Unit

No Related Items For This Description Unit

Identity Statement	
Reference Code	PT/IANTT/CC/CHANC/AFIII/040A
Level	Documento
Title	Chancelaria de D. Afonso III
Earliest Date	1274/08/01
Latest Date	1274/08/01
DU Size	2.268 liv., 1.042 mç., 34 cx., 165 proc.
Context	
Source	Torre do Tombo
Content and Structure	
Scope Content	D. Afonso III arrenda, por 600 libras (com pagamento parcelado ao longo do ano), a João Soeiro, Martinho Gonçalves e a Pedro Martins "Ventre" todos os direitos e foros que possui na Terra de Fermedo (c. de Arouca), pelo prazo de 6 anos com início no dia d
Accruals	O Arquivo da Casa da Coroa ou Arquivo Real é o primitivo núcleo da Torre do Tombo. // Nos primeiros tempos da nacionalidade a Corte não era fixa e a documentação produzida acompanhava o Rei, funcionando como o que, em termos actuais, se designaria por arquivo corrente. Os documentos mais importantes eram guardados

Figure 8.10: Descriptive Metadata for an Item in the Santa Maria da Feira Collection

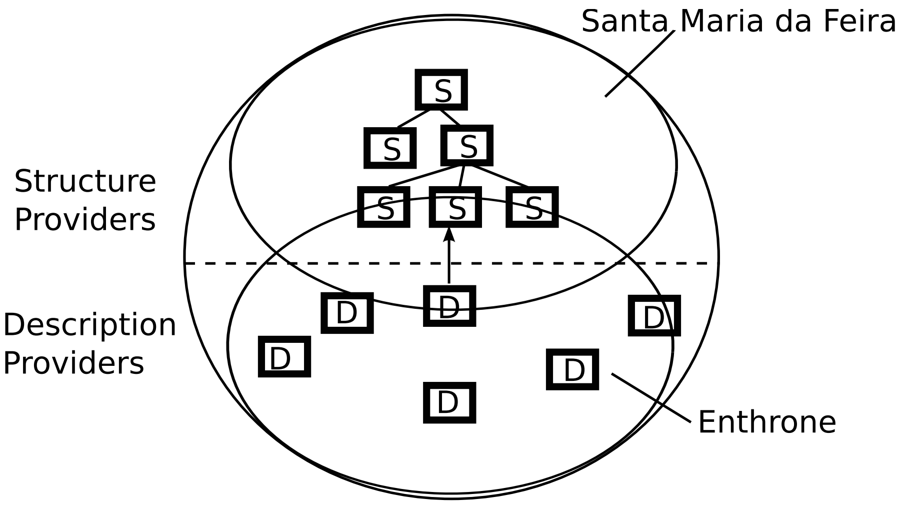


Figure 8.11: Multimedia space

The text Segments parts, i.e the transcriptions, can be searched with keyword-based queries and rely on text indexes. The results are Segments identifiers, allowing the user to quickly reach the desired content parts: digitizations and transcriptions. Figure 8.9 illustrates a "content-only" visualization of a parchment and of its transcription. The contextual metadata, stored in the DUs can be searched both with structured queries and with keyword-based queries. Visual exploration of the DUs hierarchy is also possible. These search modalities return Description Units identifiers which are used to reach "descriptive-only" visualization interfaces. Figure 8.10 illustrates the visualization of a Description Unit.

The support of multiple search modalities offers in practice a faceted retrieval of the items. Even though "structure-only", "descriptive-only" or "content-only" facets are used at a moment, the retrieval system behind them works at the level of the whole item. Whatever entry point is used, the context of the item is maintained by bringing in parallel, in separate tabs, the images, the transcriptions, the structure and the descriptions belonging to that item.

8.4 Discussion

A unified view of the multimedia items from the two case studies is shown in Figure 8.11. It can be observed that the multimedia items in the Enthrone case study have a relatively small structural component (3-4 levels) but a strong descriptive component including many types of descriptors from very diverse sources. On the other hand, the multimedia items in the Santa Maria da Feira case study have a stronger structural component (8 levels) but a descriptive component which is more uniform and comes from only two sources: one standard and one user-defined.

The two case studies presented in this Chapter helped us to empirically evaluate the required effort of implementing the MetaMedia data model in new application domains. We have identified a set of questions whose answer give insight on the appropriateness of the model. First, the application requirements must be carefully understood:

- digital items: are they simple objects, or complex documents gathering information in several modalities? What is the relationship between items and collections thereof?
- metadata: are metadata retrievable also, or just assist the content retrieval?
- standards: what metadata standards are involved?

Then, we check whether the MetaMedia data model concepts apply to the new dataset:

- is there a clear separation of content from its metadata, similar to how MetaMedia distinguishes Segments from Description Units and Descriptors?
- is it important to distinguish between context and content metadata, similar to what MetaMedia does with Description Units and Descriptors?
- is contextual metadata uniformly applied to all items?

Chapter 9

Evaluation

This chapter presents the evaluation results for MetaMedia, the retrieval system that we have proposed. The tests described here have been carried out to observe the MetaMedia's retrieval performance in the context of TREC Video Retrieval Evaluation (TRECVID), which measures the retrieval performance in a set of so-called task-oriented tests (see Section 2.8).

Although MetaMedia is built on components such as the BitMatrix index, the data model and the user interface, the results presented in this Chapter are not relevant for any of them; they reflect the performance of the retrieval system as a whole for the TRECVID set of tasks. However, component-wise evaluations have already been presented. The BitMatrix evaluation has been presented in Sections 6.3 and 6.4 and the data model and the user-interface have been successfully implemented in the case studies (Chapter 8). With positive evaluation results for the individual components, testing the whole MetaMedia system has been a natural step to be taken.

Before joining the public TRECVID benchmark evaluation [CRD^{+}07], we have conducted one independently evaluated CBIR experiment on the COREL dataset. The COREL dataset was used mostly for system-oriented tests. We remind here that Section 2.8 explains what the system-oriented and task-oriented tests are. We have tested the retrieval performance of several low-level descriptors for color and texture. Using query-by-example scenarios, the goal was to find similar images based on Color Layout, Color Structure, Scalable Color, Edge Histogram, Homogeneous Texture and Region Shape. These descriptors were extracted with the MPEG-7 XM software and indexed with BitMatrix. The ground truth was provided together with the COREL dataset and the evaluation metrics were precision and recall.

The TRECVID benchmark however, as a task-oriented evaluation, represented a much more interesting challenge. With complex query topics, a large number of participants and a common evaluation platform, TRECVID has been the most important retrieval benchmark of recent years. In the following sections we describe the TRECVID setup and the search strategy which includes aspects such as feature extraction, query topics analysis, indexing, search, and the results obtained.

9.1 The TRECVID setup

The TRECVID 2007 edition provided the context of an international video retrieval benchmark in which we could measure the MetaMedia's performances in an open, metric-based way. The development (training) and test datasets, the ground truth and evaluation measures [SOK06, Wes05] have all been provided by NIST, the organizers of TRECVID.

9.1.1 Datasets and tasks

While the previous four editions of TRECVID carried on broadcast news, the 2007 edition has accommodated its tasks on new, related, but different video genres taken from a real archive - news magazine, science news, news reports, documentaries, educational programming, and archival video.

Four tasks have been proposed:

- *shot boundary determination*: identify the shot boundaries with their location and type (cut or gradual) in the given video clip(s);
- *high-level feature extraction (automatic annotation)*: automatically annotate the video shots with concepts such as "Indoor/Outdoor", "People", "Speech". The full concept list can be found in the Appendix C;
- *search* (interactive and/or fully automatic): given the search test collection, a multimedia statement of information need (topic), and the common shot boundary reference for the search test collection, return a ranked list of at most 1000 shots from the test collection, which best satisfy the need;
- *rushes summarization*: rushes are the raw material (extra video, B-rolls footage) used to produce a video. Given a video from the rushes test collection, to automatically create an MPEG-1 summary clip less than or equal to a maximum duration (to be determined) that shows the main objects (animate and inanimate) and events in the rushes video to be summarized. The summary should minimize the number of frames used and present the information in ways that maximizes the usability of the summary and speed of objects/event recognition.

Our TRECVID participation covered mainly the search task with a small contribution to the annotation task. The datasets —common for the search and annotation tasks— consisted of 100 hours of videos from the BBC and the Danish national television, and have been divided into development and test subsets. The videos have been segmented into 36282 shots, 18140 for development and 18142 for test, and shot-wise keyframes have been extracted. In the context of the automatic annotation task, the development dataset has been manually annotated with a set of 39 LSCOM-Lite concepts [NST+06] (see Appendix C). The goal was to obtain annotated datasets that would help the development of automatic concept detectors. Such detectors have been used in the sequel to automatically annotate the test dataset on which the search tasks had to be accomplished.

9.1.2 The topics

The topics are complex multimedia statements of information need provided as bundles of 3 distinct modalities: text, image and video. The text parts of the topics, which can be consulted in the Appendix D, are natural language queries such as "Find shots of a person talking on a telephone". The other two parts are sets of image and video examples closely related to the scene described in the text part.

9.1.3 The evaluation metric

TRECVID consists of a fixed set of video shots, a fixed set of topics, a fixed set of relevance judgments and performance measures. The answer sets contain ranked lists of documents for each topic. The quality of the ranked lists is measured based on the positions of the relevant documents in the list. The average precision (AP) metric is used, where the precision is measured at every point at which a relevant document is obtained and then averaged over all relevant documents to obtain the AP value for a given topic. The score of the whole retrieval system is the mean average precision (MAP), where MAP is the mean of the AP scores for all topics (see Section 2.8).

9.1.4 The submitted runs

We have participated in the search task of TRECVID with three answer sets, also called system runs. Although our submission contained only three runs, a maximum of 6 was allowed per participant. The TRECVID statistics show that from the 24 TRECVID participants in all tasks, namely shot boundary detection, high-level feature extraction, search and rushes summarization, only 17 have participated in the search. The 17 submissions contained a total of 82 automatic search runs and 36 interactive search runs.

Of the two types of runs, the automatic runs seem more important for the MIR community due to their practical implications. It is believed that an user would rather perform several automatic searches than understand the relevance feedback mechanisms of a system. The MAP scores obtained in automatic runs are more reliable than the ones obtained with interactive ones. For example, it is known that the interactive scores can be improved not only by the relevance feedback technique itself, but by as yet uncontrollable factors such as user-interface quality, familiarization with the dataset or increased number of iterations.

9.1.5 The schedule

The target dates for the annotation and search tasks of the TRECVID 2007 edition were:

- **2. Feb** NIST sends out Call for Participation in TRECVID 2007
- **20. Feb** Applications for participation in TRECVID 2007 due at NIST
- **1. Apr** Guidelines complete
- **May-June** Download of feature/search development data

- **18. June** Download of feature/search test data
- **3. Aug** Search topics available from TRECVID website.
- **10. Aug** Feature extraction tasks submissions due at NIST for evaluation. Feature extraction donations due at NIST
- **17. Aug** Feature extraction donations available for active participants
- **20. Aug - 5. Oct** Search and feature assessment at NIST
- **10. Sep** Search task submissions due at NIST for evaluation
- **12. Oct** Results of search evaluations returned to participants
- **22. Oct** Notebook papers due at NIST
- **29. Oct** TRECVID 2007 Workshop registration closes
- **5,6 Nov** TRECVID Workshop at NIST in Gaithersburg, MD(Registration, agenda, etc)

9.2 Features

We have not proposed new descriptors for the feature extraction task. Our effort was on the efficient aggregation of the low and high-level features after having them extracted with third-party tools. In the sequel we enumerate the features that were used and the methods to obtain them.

High-level Features The high-level features come from an automatic annotation process of the TRECVID test dataset, where the annotators have been trained on a separate development set. The annotation process has been part of the feature extraction task which took place before the search task. Being a separate TRECVID task and subject to a separate evaluation process, we did not have access to a ground truth annotation, but only to the various annotation versions submitted by the participants. In the absence of a ground truth annotation we have aggregated all the participants versions and obtained our "reference annotation".

The participants in the feature extraction task have annotated the test dataset with 39 concepts from the LSCOM-Lite lexicon. Each participant has been allowed to produce up to 6 annotation versions, which resulted in a total of 163 versions. Each annotation version consisted of a list in which 2000 relevant shots for each of the concepts have been bundled. Given the 39 concepts, the length of each annotation list was at most $39 * 2000$. No degree of relevance has been present, only binary judgments: the concept was present or not in the shot. Our annotations aggregation strategy has been: for a given shot and a concept determine whether the concept was found or not in more than 30% of the lists. If the condition held, we would associate the concept to the shot.

After having obtained the reference annotation, a high-level descriptor, called "Concepts" has been created. Its construction details have already been presented in Section 7.4.

Low-level Features

Color The color feature was represented by several descriptors. We have used: “Color Layout”, “Color Structure”, “Scalable Color” and “ColorMoments”. The “ColorLayout”, “ColorStructure” and “ScalableColor” have been extracted with the MPEG-7 XM software, while the “ColorMoments” descriptor was provided by the City University of Hong Kong [JNY07].

Texture For the texture feature we have obtained “EdgeHistogram” and “Homogeneous Texture” descriptors with the MPEG-7 XM software. The “Wavelet texture” descriptor has been provided by the City University of Hong Kong and “Haralick Texture” was locally implemented based on an ImageJ version [Bai06].

Shape We have used the MPEG-7 “RegionShape” descriptor also extracted with the MPEG-7 XM reference software.

Local features Keypoints or local interest points, also known as “Scale Invariant Feature Transform” (SIFT) descriptors [Low03] have been provided by the City University of Hong Kong [JNY07].

Audio We have processed the audio tracks of the shots to obtain a set of audio features with the help of the jAudio [MMF06] software. The features are “Spectral Centroid”, “Spectral Rollof Point”, “Spectral Flux”, “Compactness”, “Spectral Variability”, “Root Mean Square”, “Fraction of Low Energy Windows”, “Zero Crossings”, “Strongest Beat”, “Beat Sum”, “Strength of Strongest Beat”, “MFCC” (Mel Frequency Cepstral Coefficient), “LPC” (Linear Predictive Coding) and “Area Method of Moments”.

9.3 Topic Processing

The TRECVID topics consisted of a natural language formulation and several image and video examples. The process of translating these topics into the bit signatures used for answer computation is described in the sequel. Figure 9.1 illustrates graphically how each topic component is translated into BitMatrix-ready signatures. The text parts of the topics have been subjected to a natural language processing (NLP) step that offered a preliminary set of TRECVID concepts, followed by a concept expansion step. The NLP step has been driven independently of the test dataset and was the result of a collaboration with a NLP group [Lab07]. The concept expansion has been applied on the set of concepts obtained after the NLP step. It had the goal of finding related concepts by analyzing the relative distances between concepts, distances that were computed specifically on the test dataset. While the NLP step has been taken independently of the dataset to be searched, the concept expansion step tried to capture the semantic “relatedness” for the particular test dataset.

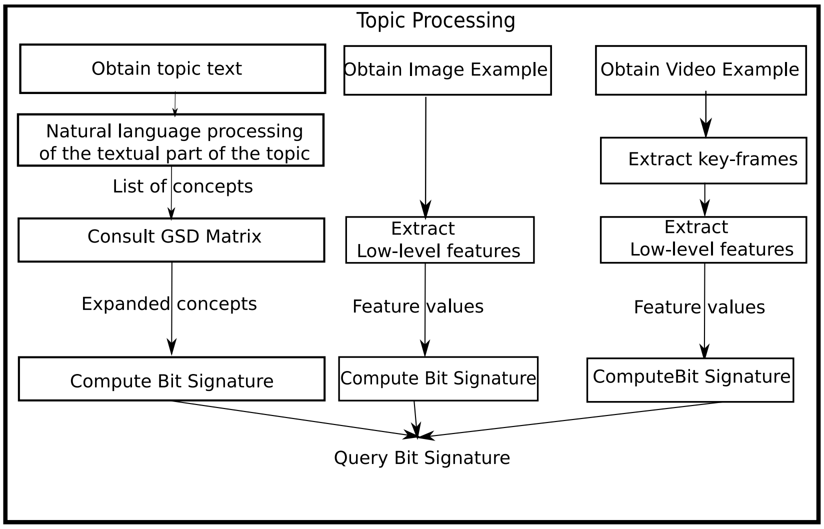


Figure 9.1: Topic Processing

9.3.1 Natural Language Processing

The goal of the NLP step was to translate the natural language queries into a set of TRECVID concepts. For example, a query such as as: “Find shots with one or more emergency vehicles in motion (e.g., ambulance, police car, fire truck, etc.)” resulted in the following list of TRECVID concepts: [car, walking, boat, disaster, airplane]. To achieve this goal we have relied on the WordNet [G. 95] lexical database and the natural language parser VISL (Visual Interactive Syntax Learning) [VIS08b, VIS08a].

Linking the TRECVID Concepts to WordNet A first step withing the NLP approach was to connect the TRECVID concepts to WordNet. WordNet is a lexical database for sets of synonyms, called synsets, with a set of semantic relations defined between them. In our approach we have used only the semantic relations between nouns, namely: hyponym (kind of), meronym (part of), instance, and synonym.

Linking the TRECVID concepts to WordNet means manually identifying for each TRECVID concept one ore more corresponding WordNet synsets. For example, the TRECVID concept “Vegetation” has been manually associated to synsets identifying concepts of botany and flora in WordNet. In Figure 9.2 the A region, illustrates the process of obtaining synsets for the TRECVID concepts.

Obtaining keywords from NL Queries The textual component of the topics have been processed in order to extract sets of keywords. This specific part is illustrated in the B region of the Figure 9.2. As an example we take the following sentence: “Find shots with one or more emergency vehicles in motion (e.g., ambulance, police car, fire truck, etc.)”. The keyword extraction process had two steps. First we have obtained the syntactic structure of the sentence using VISL

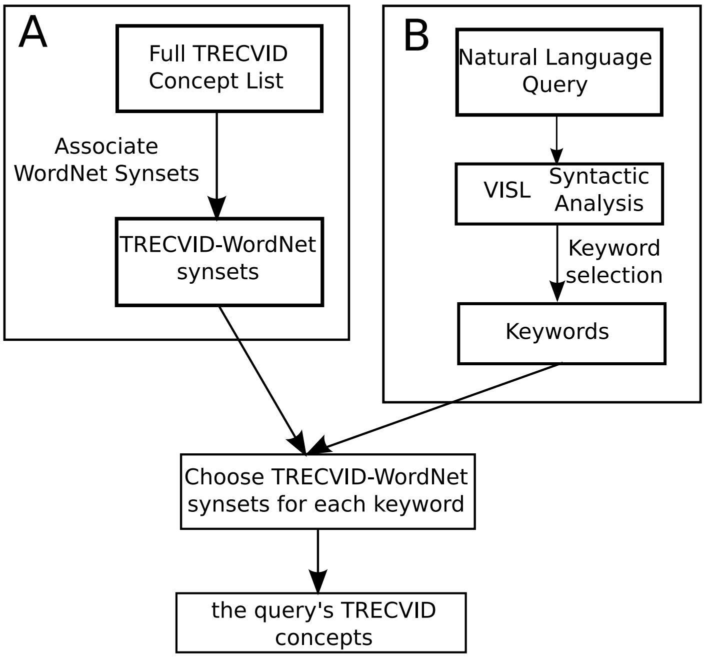

A
Full TRECVID Concept List
Associate WordNet Synsets
TRECVID-WordNet synsets
B
Natural Language Query
VISL
Syntactic Analysis
Keyword selection
Keywords
Choose TRECVID-WordNet synsets for each keyword
the query's TRECVID concepts

Figure 9.2: Processing Natural Language Queries

[VIS08b, VIS08a]. The detailed syntactic structure obtained after running VISL can be consulted in the Appendix E. Then in a *keyword selection* process, common and proper nouns have been extracted. This process looks for nouns such as “truck” or “car”, nouns preceded by a “no” such as “no clouds”, compound nouns such as “police car” and names such as “Condoleezza Rice”, that exist in WordNet. For the example sentence the result is the list: [emergency, vehicle, motion, ambulance, police car, truck].

From keywords to TRECVID concepts The last step of the NLP was to map keywords to TRECVID concepts. For each keyword we have computed the distance from its WordNet synset to all the synsets that were previously built for the TRECVID concepts. We have chosen the nearest synsets and from them the corresponding TRECVID concepts [Lab07].

9.3.2 Concept Expansion

The NLP analysis is a process based on the natural language queries and the WordNet lexical database. For a given NLP formulation, such an approach always produces a fixed set of TRECVID concepts, which is independent of the dataset to be searched. However, we felt the need of obtaining a set of concepts adapted to the test dataset. This intuition was also supported by the results in [CV07], where the authors argue that words acquire meaning from the way they are used in a specific context and their relative similarity could vary from one context to another. The authors propose the Google Similarity Distance (GSD), a similarity relation between concepts that can be quantified just by using the numbers of documents in which they occur singly and jointly. For example, a type of distance between the “horse” and “rider” terms can be computed if we know how many items contain the term “horse”, how many items contain the term “rider” and how many items contain both of the terms. Given that for the TRECVID concepts we already had a reference concept annotation (previously obtained by aggregating all the annotation versions) we could easily obtain such statistics for each pair of concepts. Thus, a dataset-dependent distance matrix containing the relative distances between the 39 TRECVID concepts has been computed using the GSD [CV07] approach. For example, we have found out that in the context of the test dataset, the “Building” concept is close to the “Court” and the “Flag-US” concepts. Therefore these concepts could also be used in queries that ask for buildings. Figure 9.1 shows that at query time, the GSD matrix is consulted after the NLP-based translation of the topics into TRECVID concepts. For each TRECVID concept in the NLP list we have added another TRECVID concept that was the closest in the GSD matrix. The final concept list is the expanded list of TRECVID concepts.

9.3.3 Examples Processing

Beside the textual part, the topics include query examples in two modalities: image and video. From them we have extracted the features enumerated in Section 9.2, in the same way we have done it for the test dataset. In the case of the image examples, the features have been directly extracted, while in the case of video examples we have identified keyframes and from them we have obtained the features. The audio features have been extracted from the audio tracks of the video examples. The middle and right columns of Figure 9.1 show the examples processing steps.

9.4 Indexing and Search

Our search strategy consisted in combining high-level features resulting from the automatic annotation task, with a large set of low-level descriptors capturing color, texture, shape, and audio features, in a way that closely follows the query-by-content method described in Section 7.4. We have combined one BitMatrix index for the high-level “Concepts” descriptor with several BitMatrix indexes for the low-level descriptors. A notable difference between our approach and some other participants approaches is the number of high-level concepts used in the search. While we have relied only on the 39 TRECVID concepts resulted from the feature extraction task, other systems have used independently built lexicons of around 1000 concepts.

First, such a strategy allowed us to independently analyze several similarity facets and evaluate experimentally each one’s contribution to the retrieval tasks. For example, the answer of a query-by-example with respect to the color feature can be obtained with several color descriptors, each one showing a different facet of color similarity. Their usefulness for the search topics had to be evaluated empirically.

Second, we have experimented with combinations of descriptors. In this case there are two important aspects to be taken in consideration. One is the quality of results and it will be detailed in the “Results” section (Section 9.5). The other is the time-efficiency. The aggregation at query time of large sets of descriptors requires efficient handling of high-dimensional feature vectors. Otherwise such a process becomes time consuming. In order to speed up the search process we handled both automatic and interactive search tasks from a database indexing perspective. At the heart of the indexing and search mechanisms has been the BitMatrix index. After having decided on the descriptors set to be used (see Section 9.2), separate BitMatrix indexes have been constructed for each one. A sample retrieval session is available on CD (Appendix G).

9.5 Results

Given the 24 topics enumerated in the Appendix D, three answer sets, called runs, have been submitted:

- *Run1*: an automatic search, without interactivity, based on color, texture and the "Concepts" descriptors;
- *Run2*: an interactive run using all the available descriptors: color, texture, shape, keypoints, audio and "Concepts";
- *Run3*: an automatic run using only textual information obtained from speech recognition followed by machine translation.

The evaluation of the automatic search *Run1* is presented in Figure 9.3, which illustrates the average precision (AP) scores. For each of the 24 topics —represented on the horizontal axis with numbers from 197 to 220— the AP are drawn as follows: the dots represent our AP values, the boxes represent the maximum AP values and the dotted line represents the median average precision. The results show that our approach performs above the median, in spite of using mostly low-level descriptors.

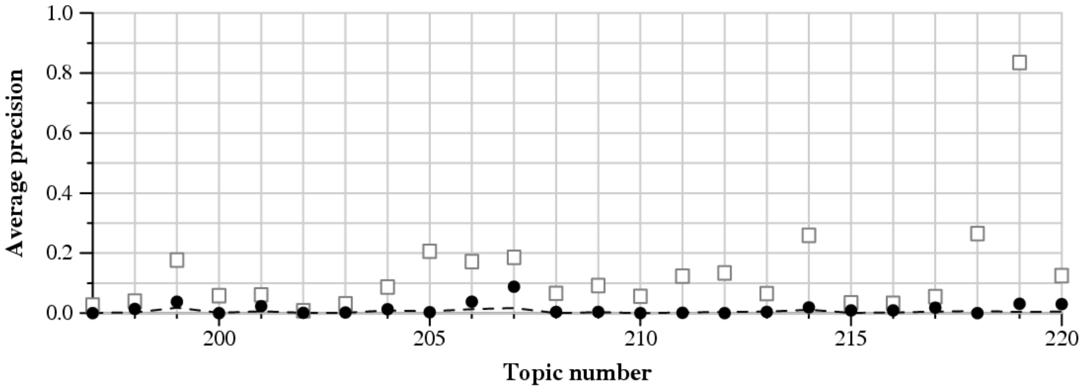


Figure 9.3: Average precision for *Run1*

The BitMatrix indexes used for automatic answer computation also support quick relevance feedback iterations. This feature allowed us to submit a second set of results, the interactive *Run2*.

All the available descriptors, enumerated in Section 9.2, have been used in this run. The AP scores of this run is presented in Figure 9.4. The significance of the dots, boxes and dotted line are same as before.

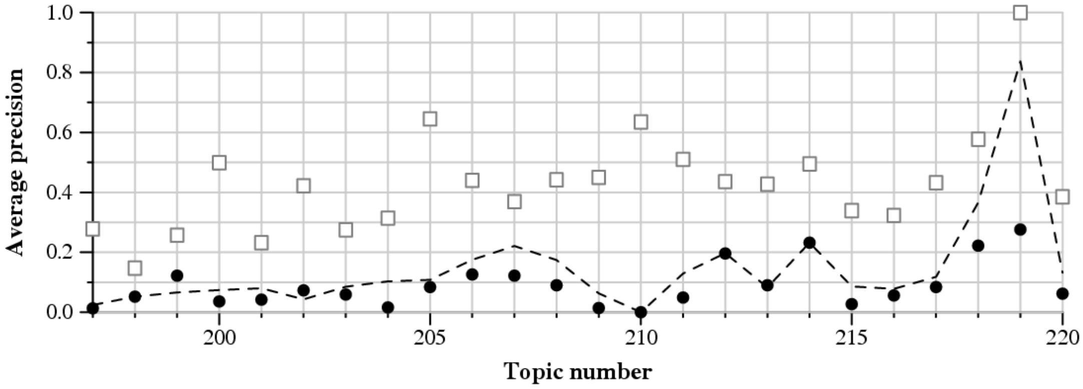


Figure 9.4: Average precision for *Run2*

The third set of results, *Run3*, has been obtained automatically. It has has been obtained by searching in the text data resulting from automatic speech recognition (ASR). As the native language used in videos was Danish, the text obtained by ASR had to be automatically translated into English (except for the native Danish speakers). The overall quality of these processes (ASR + machine translation) was quite poor. No other information such as high-level or low-level descriptors has been used. *Run3* results, illustrated in Figure 9.5, show that we have performed close to the median.

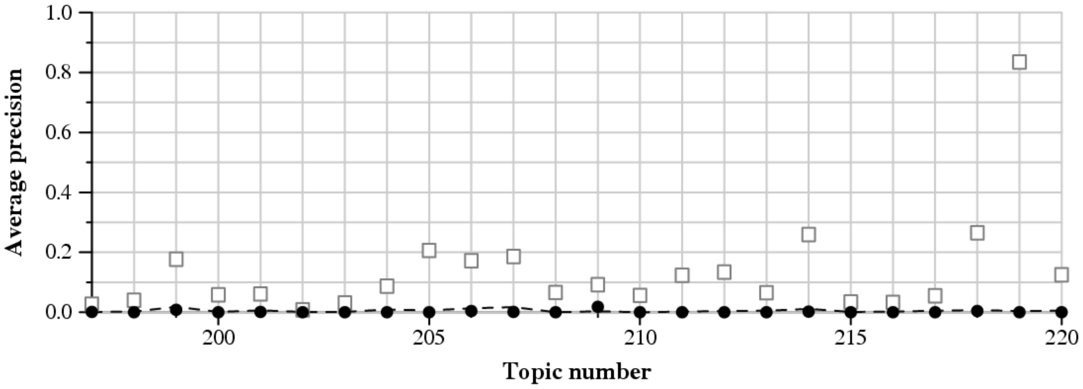
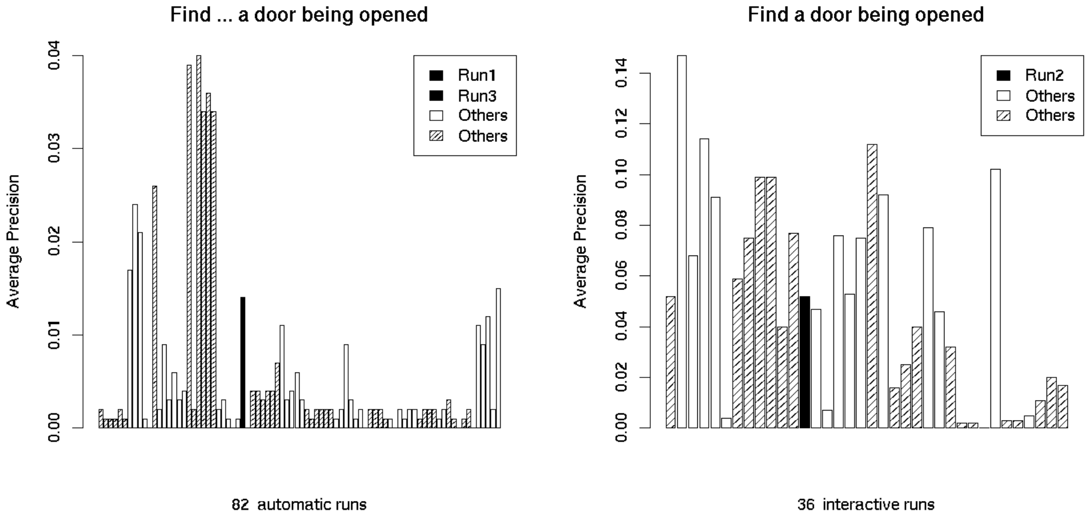


Figure 9.5: Average precision for *Run3*

A first observation regarding the previous three figures is that the performance varied significantly from topic to topic. There have been topics for which our system performed well, topics for which our scores were just above the median and topics were the performance was not satisfactory. A similar performance behavior has been observed for all the participants. That is, in Figures 9.3,



(a) Automatic

(b) Interactive

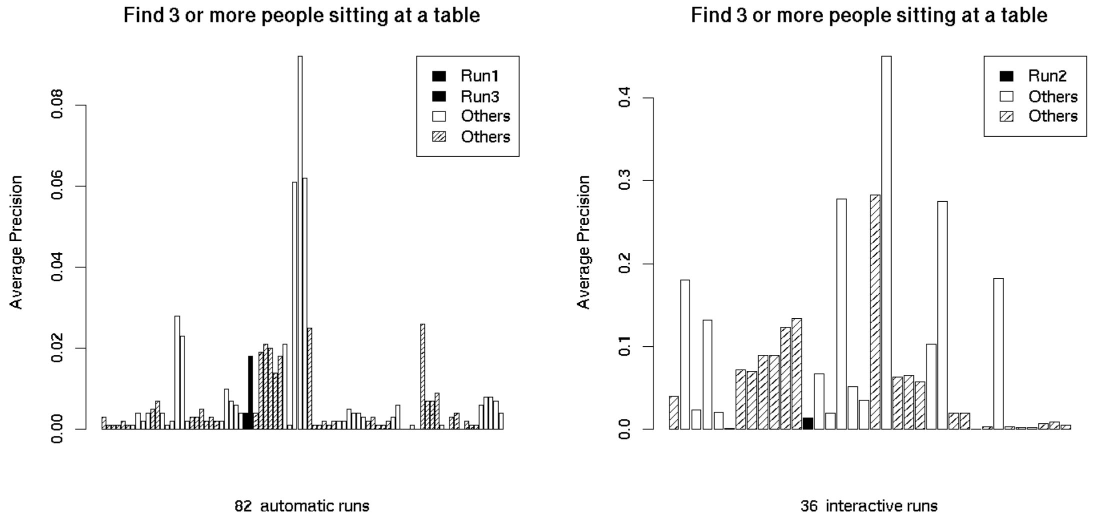
Figure 9.6: Topic 198

9.4 and 9.5 the maximum average precision (AP) scores (the boxes) were not all obtained by the same system.

A second observation is that presenting the system performances by just referring to the maximum and the median AP scores may not be informative enough. To better understand the system's behavior, we detail the results corresponding to some specific topics. The complete list of topics and their results can be found in the Appendix D and Appendix F respectively.

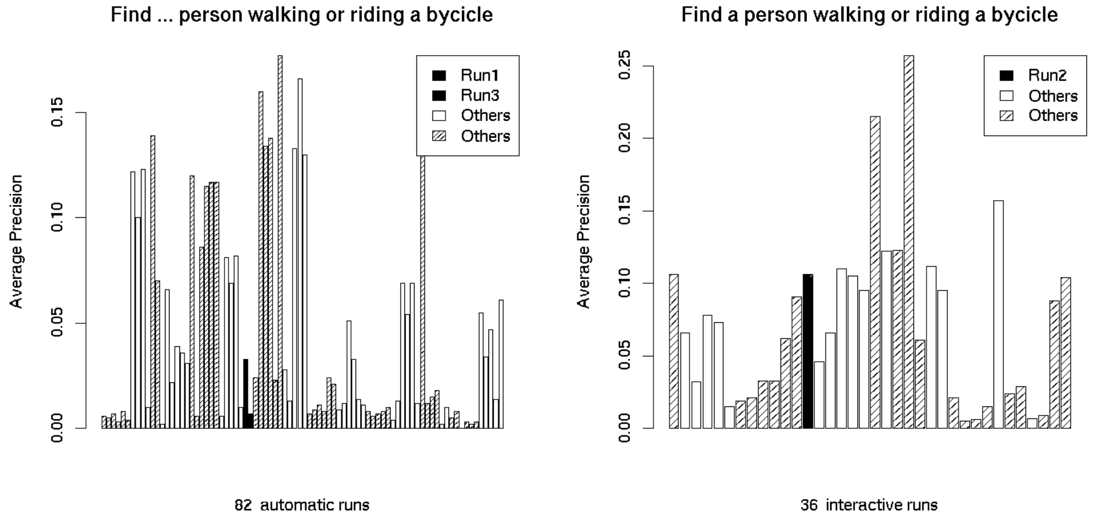
Each of the following four figures corresponds to a different topic. They present in parallel the AP scores of the automatic and the interactive runs. Each vertical bar represents a run. The left-hand sides, namely in Figures 9.6(a), 9.8(a), 9.7(a) and 9.9(a) contain the scores for all the 82 automatic runs, including our *Run1* and *Run3*. The right-hand sides of the figures, namely Figures 9.6(b), 9.8(b), 9.7(b) and 9.9(b) show the scores of the 36 interactive runs, including our *Run2*. Another detail about these figures is related to the significance of the bars. We have grouped the runs by participant (maximum 6 runs were allowed) and used alternate dashed/white type of bars for each group; our runs have solid black fill. Therefore, consecutive bars with the same fill indicate runs coming from a common participant.

Figure 9.6 illustrates the AP scores for topic number 198: *Find shots of a door being opened*. On the left, in 9.6(a) the scores for our automatic runs *Run1* and *Run3* are presented. This topic is an example where our automatic search *Run1* performed well: with an AP of 0.014 our score is the fifth from the total of 17 participants. Note that consecutive bars with the same color indicate the same participant and only the best run per participant was taken in consideration. The interactive run *Run2*, illustrated in the Figure 9.6(b) shows that the relevance feedback improved the AP score up to 0.052.



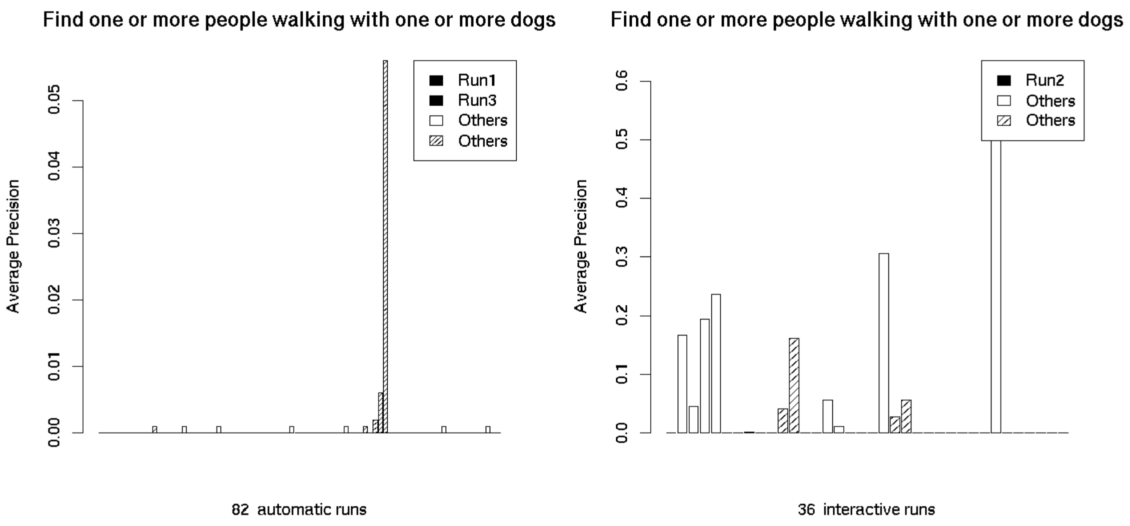
(a) Automatic (b) Interactive

Figure 9.7: Topic 209



(a) Automatic (b) Interactive

Figure 9.8: Topic 199



(a) Automatic

(b) Interactive

Figure 9.9: Topic 210

Another example is presented in Figure 9.7. This one illustrates the topic 209: *Find shots with 3 or more people sitting at a table*. This is also an example where the *Run1* scores (on the left-hand side) indicate a good performance for our automatic search. The scores for the interactive runs can also be observed in Figure 9.7(a).

Figure 9.8 illustrates the AP scores for the topic 199: *Find shots of a person walking or riding a bicycle*. This topic is an example for which the automatic score of the *Run1* (Figure 9.8(a)) was above the median. However, in this case the impact of our relevance feedback mechanism (Figure 9.8(b)) seemed more efficient comparative to the other systems.

Figure 9.9 illustrates the topic 210: *Find shots with one or more people walking with one or more dogs*. For this one, we didn't perform well neither in the automatic nor in the interactive searches.

Of the three runs that we have submitted, the AP scores of the interactive *Run2* have been generally better than the ones obtained with the automatic *Run1* and *Run3*. It is a normal behavior reflecting improvements due to relevance feedback steps. For the four topics presented above (Figures 9.6, 9.7, 9.8 and 9.9), this could be observed by comparing the AP scores in the left-hand sides with the ones on the right-hand sides. For the entire set of topics however, the AP scores of *Run1*,*Run2* and *Run3* are compared in Figure 9.10. It can be observed that even with interactivity, the AP scores of the topics 204, 208, 209 and 210 do not improve significantly. Possible explanations for this may be:

- some of the other systems had richer high-level annotations; larger sets of around 1000 concepts, covering broader semantic domains have been used.

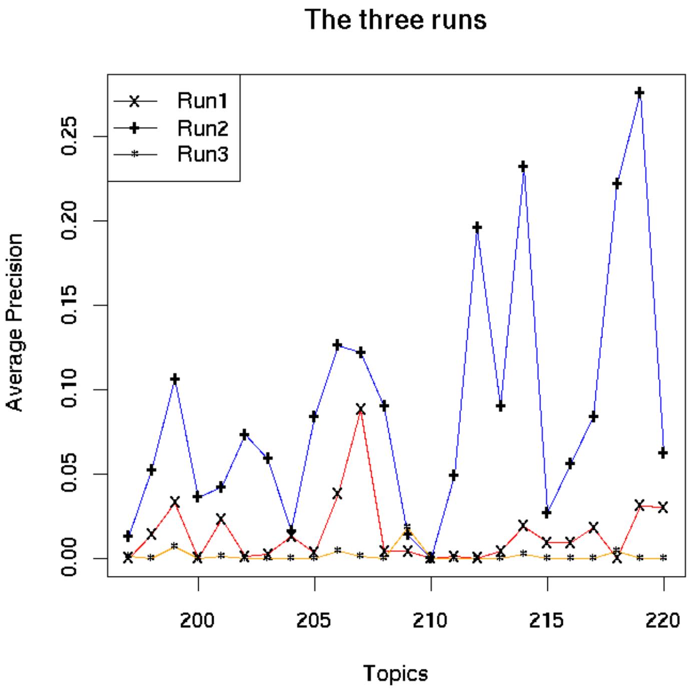


Figure 9.10: The three runs

- we have extracted features from a single representative key-frame per shot. Improvements can be obtained by using more frames for each shot.

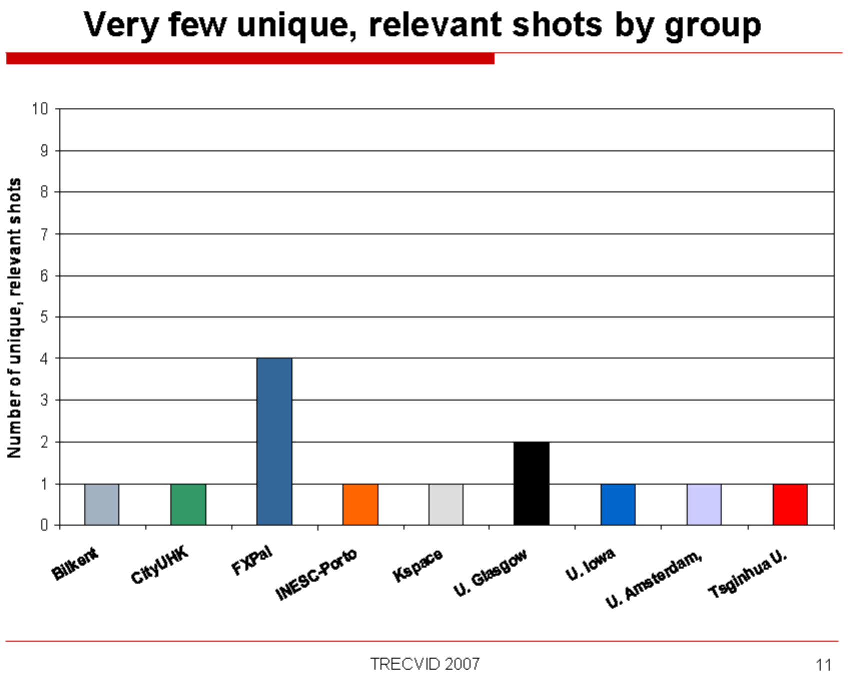
- relevance feedback was also based on key-frames but not on shot visualization.

- we have not incorporated descriptors that apply to the whole shot such as GroupOfFrames/-GroupOfPictures or MotionActivity which may capture better the nature of the shot.

- possible familiarization with the dataset, meaning that from previous experiences, one gets accustomed to the criteria, to the effect of picking specific objects, and to the weights.

A different kind of results, illustrating the diversity of the systems involved, are presented in Figure 9.11. The figure shows all the participants that found at least one unique relevant shot. The "INESC-Porto" bar in this figure illustrates that there has been a relevant shot retrieved only by our system.

Overall, the results were encouraging, given that it was our first TRECVID participation and the 2007 dataset was much more complex that the ones in the previous years. While in the previous years the data consisted mostly of news videos in English, in 2007 the data contained educational, cultural, youth-oriented programming, news magazines and historical footage videos. Beside the greater variety of subject matter, the video material has been primarily in Dutch, without repetitive parts such as commercials, repeated news footage. It has been easy to index low and high-level



Very few unique, relevant shots by group

TRECVID 2007 11

Figure 9.11: Unique shots

features with the same index type —BitMatrix— and then aggregate the separate rankings in order to experiment novel search strategies.

Chapter 10

Summary and Conclusions

Multimedia repositories support a range of applications which are essential in science, industry, security, entertainment and personal management. From surveillance for security purposes for example, to the management of personal photo archives, all this range of applications rely on functionalities such as organization and search. This work explored several aspects that can improve the management of large and heterogeneous multimedia repositories. In particular, we have studied the data organization, the search and the retrieval aspects, being motivated by the following question:

1. How to improve multimedia retrieval in large databases of multimedia items?

To answer this question we have carried a comprehensive literature review and identified specific requirements which yielded two other particular questions:

2. Given the heterogeneity of description standards, is it possible to define a common metadata model that can be applied to both collections and multimedia items?

3. High-dimensional descriptors are often used to characterize multimedia contents. Is it possible to design a versatile indexing structure for them?

For the first question we had to understand the relation between the databases and information retrieval fields. Since the early years these fields have developed independently as distinct domains. The databases have been used mostly in the so-called data retrieval, thus for precise retrieval tasks for which efficiency is the main constraint. On the other hand, dealing with the imprecise task of finding relevant documents, the information retrieval tasks are more concerned with the effectiveness (the quality of the results) than with efficiency. For instance, in the well-known TRECVID benchmark, the response times are not yet considered, although they are logged.

Nowadays, an integration effort between the databases and information retrieval fields is required. For efficiency reasons, the growth of multimedia mass production requires a shift from a simple collection paradigm to well-organized databases. At the same time, for effectiveness reasons, multimedia search problems such as the semantic gap require IR techniques. Studying how to improve multimedia retrieval in large databases of multimedia items —our first question— we have identified a list of requirements. Among them, two have been considered the most important and have raised the other two motivating questions presented above. First, at the heart of a multimedia database is the data model. Such a model should be versatile enough to support multiple multimedia applications and, in particular, should focus on capturing heterogeneous metadata. Second, the search engine of a multimedia database should be prepared for high-dimensional data. Flexible ranking mechanisms that capture user preferences, support for diverse similarity models, and efficiency, are also important requirements for a multimedia search engine.

Our answers to the second and third questions are proposed in Chapters 5 and 6. The proposed solutions are integrated in a multimedia retrieval system — MetaMedia— whose architectural and functional properties are presented in Chapter 4. We have also illustrated two real-world cases studies which are instantiations of the MetaMedia retrieval system in different domains: *Enthrone*, focused on quality of service for news broadcasting and video on demand in heterogeneous networks and the *Santa Maria da Feira* documentation center, a digital archive containing digitized documents, archival descriptions and transcription texts produced by scholars.

10.1 Data Model

In order to establish a data model that would be applicable to heterogeneous multimedia applications we have identified the main principles and from them the concepts that underlie the metadata standards in the audiovisual areas. The data models main concepts are:

Description Units, Segments and Descriptors: Our review of the current standards revealed that the multimedia items are generally hierarchically organized, possessing an amount of descriptive attributes which are uniformly meaningful at any level. We have conceptualized the set of attributes as a Description Unit, a term that comes from the standards in the archival domains. The multimedia items have been be modeled as hierarchies of Description Units. However, there are content parts of the multimedia items such as a raw video sequence or an image, that may be independently analyzed resulting in metadata that are locally applicable. We have conceptualized such content parts as Segments and the results of the Segments analysis as Descriptors.

Flexibility and control: The hierarchies of Description Units can have various topologies and different semantics for their levels. Such hierarchies can be created for new collections or can be extracted from existing ones. However, in many cases the hierarchy itself captures the nature of the datasets. Therefore, a scheme that introduced constraints on the possible levels, their semantics and their interconnections were established. We have enriched our data model with the Scheme and Scheme Level concepts.

10.1.1 Data model evaluation

The success of the two case studies shows the appropriateness of the model in domains with very different requirements, a video-on-demand platform for content distribution and a historic documentation center. The model has proved robust and the configurable hierarchy has been useful both for capturing the structure of items in a digital environment and for representing the structure of an archive where digitized versions of ancient documents are organized according to archival criteria. Other features of the model are:

- Simplicity: we kept the number of concepts, classes and attributes as small as possible.
- Implementation independent: although we have chosen a particular database system, the data model can be implemented straightforwardly in any other relational database.

10.2 Indexing High-dimensional Data

The third question addressed the problem of indexing high-dimensional data. To answer it, we have proposed the BitMatrix indexing structure. It is a highly parameterizable index structure offering a large space for experimentation: similarity threshold, number of dimensions processed in each step, and dimension processing order for the case of weighted dimensions. It also supports weighted queries and accommodates query feedback mechanism. The BitMatrix index retains most of sequential scans flexibility with good quality of the approximations and a much better time performance. It can be conveniently arranged for efficient sequential access and optimized bitwise operations. It can also be broken into segments for distributed or parallel processing.

Our experiments were driven on top of a especially developed framework for integration of high-dimensional indexing methods, called MMDI. The MMDI allowed us to test a set of 5 methods: Sequential Scan, Bond, VA-File, GridBitmap, and BitMatrix. BitMatrix has been also evaluated in an independent multimedia retrieval test [AAYK08], where it outperformed a slim-tree index.

10.3 MetaMedia

Beside the data model and the indexing structure, the MetaMedia retrieval system builds the technological structure required to incorporate them. MetaMedia supports the representation of the multimedia items and retrieval tasks such as importing and exporting items and sets of items according to common standards, browsing the repositories, and searching. The user interface is flexible enough to adapt to different kinds of repositories and allows faceted search in the structure of the items, their contents and their metadata.

10.4 Conclusions

The diversity of increasingly complex multimedia standards for both content and metadata introduces an overload of concepts and properties that must be taken in consideration by the designers of retrieval engines. Two requirements are of central importance and solutions have been proposed. The first one is the need for a data model that accommodates the wide range of standard formats and handles both text and high-dimensional numerical data. With the proposed data model, we have succeeded in identifying a set of concepts that apply to all the metadata standards that have been analyzed. However, the data model does not adapt automatically to new standards; a certain analysis step is required to map a new standard to the data model concepts. The second requirement is for a search engine that combines heterogeneous information efficiently and effectively. Putting it closer to search engines terminology, the problem is how to cope with the trade-off between flexible scoring and ranking on one side, and query execution speed on the other side. To alleviate this pressure, we have proposed the BitMatrix index which, for the sake of efficiency, prunes the majority of objects by quickly analyzing their bit signatures, while ranking the few remaining ones with any mechanism available.

10.4.1 Future work

A first work direction is towards MetaMedia improvement. Feature extraction at upload time and a better user-interface have been already identified as short term tasks. A second direction for future work includes the study of the BitMatrix adaption to the distributed and parallel paradigms. It seems to be easily adaptable to software frameworks for parallel computations over large data sets, such as “MapReduce” [DG08]. MapReduce allows to split an application among a set of machines by dividing the job into two parts: a Map, and a Reduce. We think that the BitMatrix search algorithm could also have a Map phase, which would take the initial bit matrix, split it into smaller bit matrices, and send the parts to different machines —so all would be searched at the same time. A Reduce would combine the partial results to get a single answer set. We expect that the BitMatrix adaption to “MapReduce” will allow even larger amounts of high-dimensional data to be efficiently searched.

References

[AAG02] G. Aggarwal, T.V. Ashwin, and S. Ghosal. An image retrieval system with automatic query modification. *Multimedia, IEEE Transactions on*, 4(2):201–214, Jun 2002.

[AAHH03] Grigoris Antoniou, Grigoris Antoniou, Frank Van Harmelen, and Frank Van Harmelen. Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92. Springer-Verlag, 2003.

[AAYK08] E. Acar, S. Arslan, A. Yazici, and M. Koyuncu. Slim-tree and BitMatrix index structures in image retrieval system using MPEG-7 Descriptors. *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 402–409, June 2008.

[Agg02] Charu C. Aggarwal. Towards meaningful high-dimensional nearest neighbor search by human-computer interaction. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 593–604, 2002.

[AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, 1973:420–434, 2001.

[AI01] Walid G. Aref and Ihab F. Ilyas. SP-GiST: An Extensible Database Index for Supporting Space Partitioning Trees. *J. Intell. Inf. Syst.*, 17(2-3):215–240, 2001.

[AKJ02] Sameer Antani, Rangachar Kasturi, and Ramesh Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, 2002.

[AMN$^+$98] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[AMNK02] P. de Vries Arjen, Nikos Mamoulis, Niels Nes, and Martin Kersten. Efficient k-NN search on vertically decomposed data. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 322–333. ACM Press, 2002.

[Apa05] Apache XMLBeans Home Page. http://xmlbeans.apache.org/, 2005.

[Apa06] Apache Lucene. http://lucene.apache.org/, November 2006.

[AWY99] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu. A new method for similarity indexing of market basket data. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 407–418, New York, NY, USA, 1999. ACM Press.

[AY00] Charu C. Aggarwal and Philip S. Yu. The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 119–129, New York, NY, USA, 2000. ACM Press.

[Bai06] Werner Bailer. Writing ImageJ PlugIns - A Tutorial, July 2006.

[BBK01] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, 33(3):322–373, 2001.

[BC05] Ilaria Bartolini and Paolo Ciaccia. Optimal incremental evaluation of preference queries based on ranked sub-queries. In *SEBD*, pages 308–315, 2005.

[BCJ$^+$05] Kevin Beyer, Roberta J. Cochrane, Vanja Josifovski, Jim Kleewein, George Lapis, Guy Lohman, Bob Lyle, Fatma Özcan, Hamid Pirahesh, Normen Seemann, Tuong Truong, Bert Van der Linden, Brian Vickery, and Chun Zhang. System RX: one part relational, one part XML. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 347–358, New York, NY, USA, 2005. ACM Press.

[BCOO05] Ilaria Bartolini, Paolo Ciaccia, Vincent Oria, and M. Tamer Özsu. Integrating the results of multimedia sub-queries using qualitative preferences. In *SEBD*, pages 308–315, 2005.

[BdWH$^+$03] Ian Burnett, Rik Van de Walle, Keith Hill, Jan Bormans, and Fernando Pereira. MPEG-21: Goals and Achievements. *IEEE MultiMedia*, 10(4):60–70, 2003.

[BG04] Wolf-Tilo Balke and Ulrich Güntzer. Multi-objective query processing for database systems. In *VLDB*, pages 936–947, 2004.

[BGRS99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? *Lecture Notes in Computer Science*, 1540:217–235, 1999.

[Bim99] Alberto Del Bimbo. *Visual information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[BKK96] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The X-tree: An index structure for high-dimensional data. In T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, and Nandlal L. Sarda, editors, *Proceedings of the 22nd International Conference on Very Large Databases*, pages 28–39, San Francisco, U.S.A., 1996. Morgan Kaufmann Publishers.

[BKS01a] Christian Böhm, Hans-Peter Kriegel, and Thomas Seidl. Adaptable similarity search using vector quantization. In *DaWaK '01: Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery*, pages 317–327, London, UK, 2001. Springer-Verlag.

[BKS01b] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering*, pages 421–430, Washington, DC, USA, 2001. IEEE Computer Society.

[BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19(2):322–331, 1990.

[BM72] Rudolf Bayer and Edward M. McCreight. Organization and maintenance of large ordered indices. *Acta Informatica.*, 1:173–189, 1972.

[BO99] Tolga Bozkaya and Meral Ozsoyoglu. Indexing large metric spaces for similarity search queries. *ACM Trans. Database Syst.*, 24(3):361–404, 1999.

[Bob01] M. Bober. Mpeg-7 visual shape descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):716–719, Jun 2001.

[Bul04] Dick C. A. Bulterman. Is It Time for a Moratorium on Metadata? *IEEE Multimedia*, 11(4):10–17, 2004.

[CAA$^+$07] Pedro Carvalho, Maria Andrade, Claudio Alberti, H.Castro, C. Calistru, and P. de Cuetos. A unified data model and system support for the context-aware access to multimedia content. In *Workshop of Multimedia Semantics-The Role of Metadata*, 2007.

[Cal] CalPhotos. http://calphotos.berkeley.edu/.

[CCB04] Yang Chu, Liang-Tien Chia, and Sourav S. Bhowmick. Looking at mapping, indexing & querying of MPEG-7 descriptors in RDBMS with SM3. In *MMDB '04: Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 55–64. ACM Press, 2004.

[CCB07] Yang Chu, Liang-Tien Chia, and Sourav S. Bhowmick. Mapping, indexing and querying of mpeg-7 descriptors in rdbms with ixmdb. *Data Knowl. Eng.*, 63(2):224–257, 2007.

[CDES05] Eugene Inseok Chong, Souripriya Das, George Eadon, and Jagannathan Srinivasan. An efficient sql-based rdf querying scheme. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 1216–1227. VLDB Endowment, 2005.

[CFB04] Michel Crucianu, Marin Ferecatu, and Nozha Boujemaa. Relevance feedback for image retrieval: a short review. In *In State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction including Datamodels and Languages (DELOS2 Report*, 2004.

[Cha03] Guang-Ho Cha. Bitmap indexing method for complex similarity queries with relevance feedback. In *MMDB '03: Proceedings of the 1st ACM international workshop on Multimedia databases*, pages 55–62, New York, NY, USA, 2003. ACM Press.

[Cho02] Jan Chomicki. Querying with Intrinsic Preferences. In *EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology*, pages 34–51, London, UK, 2002. Springer-Verlag.

[Cho03] Jan Chomicki. Preference formulas in relational queries. *ACM Trans. Database Syst.*, 28(4):427–466, 2003.

[CMM$^+$01] Ingemar J. Cox, Matt L. Miller, Thomas P. Minka, Thomas V. Papathomas, and Peter N. Yianilos. The bayesian image retrieval system: Pic hunter theory, implementation, and psychophysical experiments. pages 295–312, 2001.

[CNBYM01] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, 2001.

[Cod83] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 26(1):64–69, 1983.

[Con04] World Wide Web Consortium. RDF Vocabulary Description Language 1.0, consulted on 2007-03-28. `http://www.w3.org/TR/rdf-schema/`, 2004.

[CP02] Paolo Ciaccia and Marco Patella. Searching in metric spaces with user-defined and approximate distances. *ACM Trans. Database Syst.*, 27(4):398–437, 2002.

[CPZ97] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 426–435. Morgan Kaufmann Publishers Inc., 1997.

[CRD06] Catalin Calistru, Cristina Ribeiro, and Gabriel David. Multidimensional Descriptor Indexing: Exploring the BitMatrix. In Sundaram et al. [SNSR06], pages 401–410.

[CRD$^+$07] Catalin Calistru, Cristina Ribeiro, Gabriel David, Irene Rodrigues, and Gustavo Laboreiro. INESC Porto at TRECVID 2007: Automatic and Interactive Video Search, 2007.

[CRW05] Surajit Chaudhuri, Raghu Ramakrishnan, and Gerhard Weikum. Integrating db and ir technologies: What is the sound of one hand clapping? In *CIDR*, pages 1–12, 2005.

[CTB$^+$99] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

[CV07] Rudi Cilibrasi and Paul M. B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19:370, 2007.

[DB82] L. C. Dinneen and B. C. Blakesley. Definition of Spearman's footrule. *Journal of Applied Statistics*, 31(1):66–66, 1982.

[dBBD$^+$01] Jochen Van den Bercken, Bjön Blohsfeld, Jens-Peter Dittrich, Jürgen Krämer, Tobias Schäfer, Martin Schneider, and Bernhard Seeger. XXL - A Library Approach to Supporting Efficient Implementations of Advanced Database Queries. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 39–48, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[DCES04] Souripriya Das, Eugene Inseok Chong, George Eadon, and Jaannathan Srinivasan. Supporting ontology-based semantic matching in rdbms. In *vldb'2004: Proceedings of the Thirtieth international conference on Very large data bases*, pages 1054–1065. VLDB Endowment, 2004.

[DDB91] Klaus R. Dittrich, Umeshwar Dayal, and Alejandro P. Buchmann, editors. *On Object-Oriented Database Systems*. Topics in Information Systems. Springer, 1991.

[DDL$^+$90] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[DE06] Jon Ducrou and Peter Eklund. Browsing and searching mpeg-7 images using formal concept analysis. In *AIA'06: Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, pages 317–322, Anaheim, CA, USA, 2006. ACTA Press.

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[Dim04] Nevenka Dimitrova. Context and memory in multimedia content analysis. *IEEE MultiMedia*, 11(3):7–11, 2004.

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, to appear:to appear, 2008.

[DKD$^+$02] Mario Döller, Harald Kosch, Bernhard Dörflinger, Alexander Bachlechner, and Gisela Blaschke. Demonstration of an mpeg-7 multimedia data cartridge. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 85–86, New York, NY, USA, 2002. ACM.

[DKN04] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval – a quantitative comparison. In *DAGM 2004, Pattern Recognition, 26th DAGM Symposium*, LNCS, Tbingen, Germany, September 2004.

[DKNS01] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM Press.

[DN04] Ramprasath Dorairaj and K.R. Namuduri. Compact combination of MPEG-7 color and texture descriptors for image retrieval. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, volume vol.1, pages 387– 391, 2004.

[DN05] Christian Digout and Mario A. Nascimento. High-dimensional similarity searches using a metric pseudo-grid. In *ICDE Workshops 1174*, 2005.

[Dub07] Dublin Core Requirements Group. Dublin Core Metadata Initiative. `http://dublincore.org/`, 2007.

[Dum90] S. Dumais. Enhancing performance in latent semantic indexing, 1990.

[Dum91] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

[EAD07] EAD. Encoded Archival Description (EAD) Version 2002, consulted on 2007-03-08. `http://www.loc.gov/ead/`, 2007.

[EB03a] Horst Eidenberger and Christian Breiteneder. Visual similarity measurement with the feature contrast model. In Minerva M. Yeung, Rainer W. Lienhart, and Chung-Sheng Li, editors, *Proceedings SPIE Storage and Retrieval for Media Databases Conference*, volume 5021, pages 64–76. SPIE, 2003.

[EB03b] Horst Eidenberger and Christian Breiteneder. Vizir–a framework for visual information retrieval. *Journal of Visual Languages & Computing*, 14(5):443–469, October 2003.

[Eid02] C. Eidenberger, H.; Breiteneder. An experimental study on the performance of visual information retrieval similarity models. *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 233–236, 9-11 Dec. 2002.

[Eid03] Horst Eidenberger. Distance measures for MPEG-7-based retrieval. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 130–137, New York, NY, USA, 2003. ACM Press.

[ETM04] Ahmet Ekin, A. Murat Tekalp, and Rajiv Mehrotra. Integrated Semantic-Syntactic Video Modeling for Search and Browsing. *IEEE Transactions on Multimedia*, 6(6):839–851, 2004.

[FA] Herwig Lejsek Fridrik Ásmunnsson. The application of the medrank algorithm to content-based image retrieval using local descriptors.

[Fag99] Ronald Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[Fag02] Ronald Fagin. Combining fuzzy information: an overview. *SIGMOD Rec.*, 31(2):109–118, 2002.

[FFG89] Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The structure-mapping engine: algorithm and examples. *Artif. Intell.*, 41(1):1–63, 1989.

[FI92] Usama M. Fayyad and Keki B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Mach. Learn.*, 8(1):87–102, 1992.

[FKS03] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM Press.

[FL95] Christos Faloutsos and King-Ip Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174, New York, NY, USA, 1995. ACM.

[FO95] Christos Faloutsos and Douglas W. Oard. A survey of information retrieval and filtering methods. Technical report, Univ. of Maryland Institute for Advanced Computer Studies Report, College Park, MD, USA, 1995.

[Fol90] P. W. Foltz. Using latent semantic indexing for information filtering. *SIGOIS Bull.*, 11(2-3):40–47, 1990.

[Fou] Apache Software Foundation. Jetspeed. `http://jakarta.apache.org/jetspeed/site/index.html`.

[FS04] Thorsten Fiebig and Harald Schöning. Software AG's Tamino XQuery Processor. In *XIME-P*, pages 19–24, 2004.

[FSN$^+$95] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathon Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, 1995.

[FTCTF01] Roberto F. Santos Filho, Agma J. M. Traina, Jr. Caetano Traina, and Christos Faloutsos. Similarity search without tears: The omni family of all-purpose access methods. In *Proceedings of the 17th International Conference on Data Engineering*, pages 623–630, Washington, DC, USA, 2001. IEEE Computer Society.

[G. 95] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, pages 39–45, November 1995.

[GBK00] Ulrich Güntzer, Wolf-Tilo Balke, and Werner Kießling. Optimizing multi-feature queries for image databases. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 419–428. Morgan Kaufmann, 2000.

[GCRD07] Bruno Gonçalves, Catalin Calistru, Cristina Ribeiro, and Gabriel David. An Evaluation Framework for Multidimensional Multimedia Descriptor Indexing. In *Workshop on Multimedia Databases and Data Management*, 2007.

[Gen88] D. Gentner. Structure-mapping: A theoretical framework for analogy. In A. Collins and E. E. Smith, editors, *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pages 303–310. Kaufmann, San Mateo, CA, 1988.

[GJ97] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, 1997.

[GPB04] Jonathan Goldstein, John C. Platt, and Christopher J. C. Burges. Redundant Bit Vectors for Quickly Searching High-Dimensional Regions. In *Deterministic and Statistical Methods in Machine Learning*, pages 137–158, 2004.

[GR95] Venkat N. Gudivada and Vijay V. Raghavan. Content-based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[GS04] Theo Gevers and A. W. M. Smeulders. *Content-Based Image Retrieval: An Overview*. IMSC Press Multimedia Series. Prentice Hall, 1st edition, July 2004.

[GSG05] Parke Godfrey, Ryan Shipley, and Jarek Gryz. Maximal vector computation in large data sets. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 229–240. VLDB Endowment, 2005.

[Had] Marios Hadjieleftheriou. R-tree visualization. `http://www.dbnet.ece.ntua.gr/~mario/rtree/`.

[HAK00] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What Is the Nearest Neighbor in High Dimensional Spaces? In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[Han06] A. Hanjalic. Extracting moods from pictures and sounds: towards truly personalized tv. *Signal Processing Magazine, IEEE*, 23(2):90–100, 2006.

[HC04] Alexander G. Hauptmann and Michael G. Christel. Successful approaches in the trec video retrieval evaluations. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 668–675, New York, NY, USA, 2004. ACM.

[HK] Nick Hawes and John Kelleher. Analogy by alignment: On structure mapping and similarity.

[HL05] Alexander G. Hauptmann and Wei-Hao Lin. Assessing effectiveness in video retrieval. In *CIVR*, pages 215–225, 2005.

[HLJ06] Steven C.H. Hoi, Michael R. Lyu, and Rong Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):509–524, 2006.

[HNP95] Joseph M. Hellerstein, Jeffrey F. Naughton, and Avi Pfeffer. Generalized search trees for database systems. In *VLDB*, pages 562–573, 1995.

[HOdJ07] Marijn Huijbregts, Roeland Ordelman, and Franciska de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of the second international conference on Semantics And digital Media Technologies (SAMT)*, Lecture Notes in Computer Science, Berlin, December 2007. Springer Verlag.

[H.R91] H.R.Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University of Massachusetts, 1991.

[HR05a] P. Howarth and S. Rüger. Robust texture features for still-image retrieval. *Vision, Image and Signal Processing, IEE Proceedings* -, 152(6):868–874, 9 Dec. 2005.

[HR05b] Peter Howarth and Stefan Rüger. *Image and Video Retrieval*, volume `http://www.springerlink.com/content/eq86j0456l73ax78`, chapter Trading Precision for Speed: Localised Similarity Functions, pages 415–424. Springer Berlin / Heidelberg, 2005.

[HR05c] Peter Howarth and Stefan M. Rüger. Fractional distance measures for content-based image retrieval. In *ECIR*, pages 447–456, 2005.

[HS03] Gisli R. Hjaltason and Hanan Samet. Index-driven similarity search in metric spaces. *ACM Trans. Database Syst.*, 28(4):517–580, 2003.

[HW97] Alexander G. Hauptmann and Michael J. Witbrock. *Informedia: news-on-demand multimedia information acquisition and retrieval*, pages 215–239. MIT Press, Cambridge, MA, USA, 1997.

[HYL07] Alexander Hauptmann, Rong Yan, and Wei-Hao Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634, New York, NY, USA, 2007. ACM.

[IA03] Qasim Iqbal and J.K. Aggarwal. Feature integration, multi-image queries and relevance feedback in image retrieval, 2003.

[IBM08] IBM. Marvel: Multimedia Analysis and Retrieval. `http://mp7.watson.ibm.com/marvel/`, January 2008.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of 30th STOC*, pages 604–613, 1998.

[ISA99] ISAD(G). General International Standard Archival Description, Second edition. `http://www.ica.org/`, 1999.

[ISA04] ISAAR(CPF). International Standard Archival Authority Record for Corporate Bodies, Persons, and Families, Second edition. `http://www.ica.org/`, 2004.

[ISF98] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 218–227, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[Jag06] Jagadish, H.V. and Beng Chin Ooi and Heng Tao Shen and Kian-Lee Tan. Toward Efficient Multifeature Query Processing. *Knowledge and Data Engineering, IEEE Transactions on*, 18(03):350–362, 2006.

[JAKC$^+$02] H. V. Jagadish, S. Al-Khalifa, A. Chapman, L. V. S. Lakshmanan, A. Nierman, S. Paparizos, J. M. Patel, D. Srivastava, N. Wiwatwattana, Y. Wu, and C. Yu. Timber: A native xml database. *The VLDB Journal*, 11(4):274–291, 2002.

[JCLZ05] Wei Jiang, Kap Luk Chan, Mingjing Li, and Hongjiang Zhang. Mapping low-level features to high-level semantic concepts in region-based image retrieval. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 244–249, Washington, DC, USA, 2005. IEEE Computer Society.

[JD01] S. Jeannin and A. Divakaran. Mpeg-7 visual motion descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):720–724, 2001.

[JNY07] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In Nicu Sebe and Marcel Worring, editors, *CIVR*, pages 494–501. ACM, 2007.

[Jol86] I. T. Jolliffe. *Principal component analysis*. Springer-Verlag, 1986.

[JOT+05] H. V. Jagadish, Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Rui Zhang. iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.*, 30(2):364–397, 2005.

[JUn05] JUnit Home Page. `http://www.junit.org/`, 2005.

[KBD+05] Harald Kosch, Laszlo Boszormenyi, Mario Doller, Mulugeta Libsie, Peter Schojer, and Andrea Kofler. The Life Cycle of Multimedia Metadata. *IEEE Multimedia*, 12(1):80–86, 2005.

[KBT05] Charles Kemp, Aaron Bernstein, and Joshua B. Tenenbaum. A generative theory of similarity. In *The Twenty-Seventh Annual Conference of the Cognitive Science Society*, 2005.

[KIR+01] Ho Kyung, Kang Information, Yong Man Ro, Yong Man Ro, Munchurl Kim, Munchurl Kim, Ho Kyung Kang, B. S. Manjunath, and Jinwoong Kim. Mpeg-7 homogeneous texture descriptor. *ETRI Journal*, 23(23):41–51, 2001.

[Kos03] Harald Kosch. *Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21.* CRC Press,, 2003.

[Kra05] Wessel Kraaij. Variations on language modeling for information retrieval. *SIGIR Forum*, 39(1):61, 2005.

[KRR02] Donald Kossmann, Frank Ramsak, and Steffen Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *VLDB*, pages 275–286, 2002.

[KS01] Norio Katayama and Shin'ichi Satoh. Distinctiveness-Sensitive Nearest Neighbor Search for Efficient Similarity Retrieval of Multimedia Information. In *Proceedings of the 17th International Conference on Data Engineering*, pages 493–502, Washington, DC, USA, 2001. IEEE Computer Society.

[KZB04] M. L. Kherfi, D. Ziou, and A. Bernardi. Image retrieval from the world wide web: Issues, techniques, and systems. *ACM Comput. Surv.*, 36(1):35–67, 2004.

[Lab07] Gustavo Laboreiro. Natural language classifier. Master's thesis, Evora University, Evora, November 2007.

[Lar08] Large-Scale Concept Ontology for Multimedia. `http://www.lscom.org//`, 2008.

[LB97] Todd A. Letsche and Michael W. Berry. Large-scale information retrieval with latent semantic indexing. *Information Sciences*, 100(1-4):105–137, 1997.

[LH04] Suzanne Little and Jane Hunter. Rules-by-example - a novel approach to semantic indexing and querying of images. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 534–548. Springer, 2004.

[LK02] Achim Leubner and Werner Kießling. Personalized keyword search with partial-order preferences. In *SBBD*, pages 181–193, 2002.

[LLL01] Swanwa Liao, Mario A. Lopez, and Scott T. Leutenegger. High dimensional similarity search with space filling curves. In *Proceedings of the 17th International Conference on Data Engineering*, pages 615–622, Washington, DC, USA, 2001. IEEE Computer Society.

[LMO$^+$96] Ray R. Larson, Jerome McDonough, Paul O'Leary, Lucy Kuntz, and Ralph Moon. Cheshire II: designing a next-generation online catalog. *J. Am. Soc. Inf. Sci.*, 47(7):555–567, 1996.

[Low03] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[LSDJ06] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[MFKMT$^+$00] A. Mufit Ferman, S. Krishnamachari, A. Murat Tekalp, M. Abdel-Mottaleb, and R. Mehrotra. Group-of-frames/pictures color histogram descriptors for multimedia applications. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 1:65–68 vol.1, 2000.

[MHH97] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Towards a multimedia world-wide web information retrieval engine. In *Selected papers from the sixth international conference on World Wide Web*, pages 1181–1191, Essex, UK, 1997. Elsevier Science Publishers Ltd.

[MHH99] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Amore: A world wide web image retrieval engine. *World Wide Web*, 2(3):115–132, 1999.

[MKS] Pavel Moravec, Michal Kolovrat, and Vaclav Snasel. LSI vs. Wordnet Ontology in Dimension Reduction for Information Retrieval.

[ML] Nicholas Möenne-Loccoz. High dimensional access methods for efficient similarity queries.

[MM96] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[MM99] Wei-Ying Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[MMF06] Daniel McEnnis, Cory McKay, and Ichiro Fujinaga. jaudio: Additions and improvements. In *ISMIR*, pages 385–386, 2006.

[MO06] Craig Macdonald and Iadh Ounis. Searching for expertise using the terrier platform. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 732–732, New York, NY, USA, 2006. ACM.

[Moj04] Aleksandra Mojsilovic. Semantic Metric for Image Library Exploration. *IEEE Transactions on Multimedia*, 6(6):828–838, 2004.

[MOVY01] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, 2001.

[MPE02] MPEG-21 Requirements Group. MPEG-21 Overview v.5, consulted on 2007-03-28. `http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm`, 2002.

[MPE04] MPEG-21 Requirements Group. Multimedia Framework (MPEG-21) - Part 2: Digital Item Declaration, July 2004.

[Nac04] Frank Nack. The Future in Digital Media Computing is Meta. *IEEE Multimedia*, 11(2):10–13, 2004.

[NC06] Philip H. P. Nguyen and Dan Corbett. A basic mathematical framework for conceptual graphs. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):261–271, 2006.

[New07] NewsML Requirements Group. NewsML. `http://www.newsml.org/`, 2007.

[NL99] Frank Nack and Adam T. Lindsay. Everything You Wanted to Know About MPEG-7: Part 2. *IEEE Multimedia*, 6(4):64–73, 1999.

[NMH02] Munehiro Nakazato, Lubomir Manola, and Thomas S. Huang. Imagegrouper: Search, annotate and organize images by groups. In *VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, pages 129–142, London, UK, 2002. Springer-Verlag.

[NNT05] Apostol (Paul) Natsev, Milind R. Naphade, and Jelena TešiĆ. Learning the semantics of multimedia queries and concepts from a small number of examples. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 598–607, New York, NY, USA, 2005. ACM Press.

[NST$^+$06] Milind Naphade, John R. Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. Large-Scale Concept Ontology for Multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[NvOH05] Frank Nack, Jacco van Ossenbruggen, and Lynda Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web, Part 2. *IEEE MultiMedia*, 12(1):54–63, 2005.

[OLIG04] Paul Over, Clement Leung, Horace Ip, and Michael Grubinger. Multimedia retrieval benchmarks. *IEEE MultiMedia*, 11(2):80–84, 2004.

[Ope05] Open Source Libraries for High Performance Scientific and Technical Computing in JavaWorld. `http://hoschek.home.cern.ch/hoschek/colt/`, 2005.

[Ora04] Oracle XML Technology Center. `http://otn.oracle.com/tech/xml/index.html`, January 2004.

[Ora08] Oracle Technology Center. `http://www.oracle.com/database/index.html`, February 2008.

[ORC^{+}97] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in MARS. In *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, pages 403–413, New York, NY, USA, 1997. ACM Press.

[OTYB00] Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Stéphane Bressan. Indexing the edges - a simple and yet efficient approach to high-dimensional indexing. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA*, pages 166–174. ACM, 2000.

[PBG04] B. G. Prasad, K. K. Biswas, and S. K. Gupta. Region-based image retrieval using integrated color, shape, and location index. *Comput. Vis. Image Underst.*, 94(1-3):193–233, 2004.

[PC98] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM.

[PJET05] Jian Pei, Wen Jin, Martin Ester, and Yufei Tao. Catching the best views of skyline: a semantic approach based on decisive subspaces. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 253–264. VLDB Endowment, 2005.

[Po05] Ka-Man Wong; Kwok-Wai Cheung; Lai-Man Po. Mirror: an interactive content based image retrieval system. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1541–1544 Vol. 2, 23-26 May 2005.

[PPS96] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *Int. J. Comput. Vision*, 18(3):233–254, 1996.

[PTFS05] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1):41–82, 2005.

[Rau04] M. Raubal. Formalizing conceptual spaces. In *FOIS 2004: Proceedings of the Third International Conference*, pages 153–164, A. Varzi and L. Vieu. Amsterdam, NL, 2004. IOS Press.

[RCB05] Julien Ricard, David Coeurjolly, and Atilla Baskurt. Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recogn. Lett.*, 26(14):2174–2186, 2005.

[RD01] Cristina Ribeiro and Gabriel David. Metadata Model for Multimedia Databases. In *Proceedings of the International Cultural Heritage Informatics Meeting, ICHIM01*, 2001.

[RDB06] Cristina Ribeiro, Gabriel David, and Andre Barbosa. XML Annotation of Historic Documents for Automatic Indexing. In *XATA2006, XML: Aplicaï¿$\frac{1}{2}$ï¿$\frac{1}{2}$es e Tecnologias Associadas*, pages 325–336, Portalegre, Portugal, 2006. Universidade do Minho.

[RDC04] Cristina Ribeiro, Gabriel David, and Catalin Calistru. A Multimedia Database Workbench for Content and Context Retrieval. In *MultMSP IEEE Workshop*. IEEE Computer Society Press, 2004.

[RDC07a] Cristina Ribeiro, Gabriel David, and Catalin Calistru. A historic documentation repository for specialized and public access. In László Kovács, Norbert Fuhr, and Carlo Meghini, editors, *ECDL*, volume 4675 of *Lecture Notes in Computer Science*, pages 555–558. Springer, 2007.

[RDC07b] Cristina Ribeiro, Gabriel David, and Catalin Calistru. Multimedia in cultural heritage collections: A model and applications. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Sølvberg, and Edie M. Rasmussen, editors, *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 186–195. Springer, 2007.

[RE03] M. Andrea Rodriguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, 2003.

[RMBB89] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

[RNS02] M. V. Ramakrishna, S. Nepal, and P. K. Srivastava. A heuristic for combining fuzzy results in multimedia databases. In *Proceedings of the thirteenth Australasian conference on Database technologies*, pages 141–144. Australian Computer Society, Inc., 2002.

[Rob97] S. E. Robertson. *The probability ranking principle in IR*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[RSB$^+$04] Herwig Rehatschek, Peter Schallauer, Werner Bailer, Werner Haas, and Alfred Wertner. An innovative system for formulating complex, combined content-based and keyword based queries. In S. Santini and R. Schettini, editors, *Proceedings of the SPIE conference on Internet Imaging*, pages 160 – 169, 2004.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000.

[RTL02] J. Wenny Rahayu, David Taniar, and Xiaoyan Lu. Aggregation query model for oodbms. In *Proceedings of the Fortieth International Confernece on Tools Pacific*, pages 143–150. Australian Computer Society, Inc., 2002.

[RVM06] Nikhil Rasiwasia, Nuno Vasconcelos, and Pedro J. Moreno. Query by semantic example. In Sundaram et al. [SNSR06], pages 51–60.

[Sal71] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[Sal89] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[SBD06] Ben Shneiderman, Benjamin B. Bederson, and Steven M. Drucker. Find that photo!: interface strategies to annotate, browse, and share. *Commun. ACM*, 49(4):69–71, 2006.

[SC96] John R. Smith and Shih-Fu Chang. VisualSEEk: A Fully Automated Content-Based Image Query System. In *ACM Multimedia*, pages 87–98, 1996.

[Sch05] Angela Schwering. Hybrid model for semantic similarity measurement. In Robert Meersman, Zahir Tari, Mohand-Said Hacid, John Mylopoulos, Barbara Pernici, Özalp Babaoglu, Hans-Arno Jacobsen, Joseph P. Loyall, Michael Kifer, and Stefano Spaccapietra, editors, *OTM Conferences (2)*, volume 3761 of *Lecture Notes in Computer Science*, pages 1449–1465. Springer, 2005.

[SfC97] John R. Smith and Shih fu Chang. *Querying by color regions using VisualSEEk content-based visual query system*, pages 23–41. MIT Press, Cambridge, MA, USA, 1997.

[SJ97] Simone Santini and Ramesh Jain. Similarity is a Geometer. *Multimedia Tools Appl.*, 5(3):277–306, 1997.

[SJ99] Simone Santini and Ramesh Jain. Similarity measures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):871–883, 1999.

[SKP02] José María Martínez Sanchez, Rob Koenen, and Fernando Pereira. MPEG-7: The Generic Multimedia Content Description Standard, Part 1. *IEEE MultiMedia*, 9(2):78–87, 2002.

[SLZ$^+$03] N. Sebe, M. Lew, X. Zhou, T. Huang, and E. Bakker. The State of the Art in Image and Video Retrieval. *Lecture Notes in Computer Science 2728 (Image and Video Retrieval)*, 2003.

[SM95] Michael Stonebraker and Dorothy Moore. *Object Relational DBMSs: The Next Great Wave*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[SNSR06] Hari Sundaram, Milind R. Naphade, John R. Smith, and Yong Rui, editors. *Image and Video Retrieval, 5th International Conference, CIVR 2006, Tempe, AZ, USA, July 13-15, 2006, Proceedings*, volume 4071 of *Lecture Notes in Computer Science*. Springer, 2006.

[SOK04] Alan F. Smeaton, Paul Over, and Wessel Kraaij. TRECVID: evaluating the effectiveness of information retrieval tasks on digital video. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 652–655. ACM Press, 2004.

[SOK06] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVID. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[SS06] John R. Smith and Peter Schirling. Metadata standards roundup. *IEEE MultiMedia*, 13(2):84–88, 2006.

[SW05] Cees G. M. Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools Appl.*, 25(1):5–35, 2005.

[SWdR$^+$08] Cees G. M. Snoek, Marcel Worring, Ork de Rooij, Koen E.A. van de Sande, Rong Yan, and Alexander G. Hauptmann. VideOlympics: Real–time evaluation of multimedia retrieval systems. *IEEE MultiMedia*, 15(1), January–March 2008.

[SWG$^+$06] C. G. M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, F.J. Seinstra, and A.W.M. Smeulders. The Semantic Pathfinder: Using an Authoring Metaphor for Generic Multimedia Indexing. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(10):1678–1689, 2006.

[SWRS06] M.C. Schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mSpace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, 49(4):47–49, 2006.

[SWS$^+$00] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[SWvG$^+$06] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM.

[SZ05] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA, 2005. ACM Press.

[SZZ07] Heng Tao Shen, Xiaofang Zhou, and Aoying Zhou. An adaptive and dynamic dimensionality reduction method for high-dimensional indexing. *The VLDB Journal*, 16(2):219–234, 2007.

[TBM03] J. Tesic, S. Bhagavathy, and B. S. Manjunath. Issues concerning dimensionality and similarity search. In *3rd International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep 2003.

[TEO01] Kian-Lee Tan, Pin-Kwang Eng, and Beng Chin Ooi. Efficient progressive skyline computation. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 301–310, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Tes04] Jelena Tesic. *Managing Large-scale Multimedia Repositories*. PhD thesis, University of California, Santa Barbara, Sep 2004.

[The] The ENTHRONE Project. `http://enthrone.org/`.

[TLS04] Belle L. Tseng, Ching-Yung Lin, and John R. Smith. Using MPEG-7 and MPEG-21 for Personalizing Video. *IEEE Multimedia*, 11(1):42–53, 2004.

[TM03] Jelena Tesic and B.S. Manjunath. Nearest neighbor search for relevance feedback. *cvpr*, 02:643, 2003.

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8(6):460 – 473, 1978.

[TS92] Jean Tague-Sutcliffe. The pragmatics of information retrieval experimentation, revisited. *Inf. Process. Manage.*, 28(4):467–490, 1992.

[TV-07] TV-Anytime Requirements Group. `http://www.tv-anytime.org/`, 2007.

[Tve77] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[UJ04] Jana Urban and Joemon M. Jose. Evidence combination for multi-point query learning in content-based image retrieval. In *ISMSE '04: Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04)*, pages 583–586, Washington, DC, USA, 2004. IEEE Computer Society.

[VIS08a] VISL: web application. `http://beta.visl.sdu.dk/visl/en/parsing/automatic/parse.php`, February 2008. consulted in 2008-02-22.

[VIS08b] VISL: website. `http://visl.sdu.dk/`, February 2008. consulted in 2008-02-22.

[vONH04] Jacco van Ossenbruggen, Frank Nack, and Lynda Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1. *IEEE MultiMedia*, 11(4):38–48, 2004.

[VR79] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[WB04] Li Wu and Timo Bretschneider. Vp-emd tree: An efficient indexing strategy for image retrieval. In Hamid R. Arabnia, editor, *CISST*, pages 421–426. CSREA Press, 2004.

[Wei07] Gerhard Weikum. DB&IR: both sides now. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 25–30, New York, NY, USA, 2007. ACM.

[Wes04] T. Westerveld. *Using generative probabilistic models for multimedia retrieval.* Ph.d. thesis, University of Twente, Enschede, The Netherlands, November 2004.

[Wes05] T. Westerveld. TRECVID as a Re-Usable Test-Collection for Video Retrieval. In *Proceedings of the Multimedia Information Retrieval Workshop 2005*, 2005.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

[WJ96] David A. White and Ramesh Jain. Similarity indexing with the ss-tree. In *ICDE '96: Proceedings of the Twelfth International Conference on Data Engineering*, pages 516–523, Washington, DC, USA, 1996. IEEE Computer Society.

[WK03] Utz Westermann and Wolfgang Klas. An analysis of XML database solutions for the management of MPEG-7 media descriptions. *ACM Comput. Surv.*, 35(4):331–373, 2003.

[WKSS96] Howard D. Wactlar, Takeo Kanade, Michael A. Smith, and Scott M. Stevens. Intelligent access to digital video: Informedia project. *Computer*, 29(5):46–52, 1996.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 194–205, 24–27 1998.

[WvGC$^+$05] T. Westerveld, J. C. van Gemert, R. Cornacchia, D. Hiemstra, and A. P. de Vries. An integrated approach to text and image retrieval. In *Trec Video Retrieval Evaluation Online Proceedings*, 2005.

[WY06] Quan Wang and Suya You. Fast similarity search for high-dimensional dataset. In *ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia*, pages 799–804, Washington, DC, USA, 2006. IEEE Computer Society.

[XML07] XML. `http://www.w3.org/XML/`, January 2007.

[XZTT06] Ziyou Xiong, Xiang Sean Zhou, Qi Tian, and Yong Ruiand Huangm TS. Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports. *Signal Processing Magazine, IEEE*, 23(2):18–27, 2006.

[YBOT04] Cui Yu, Stéphane Bressan, Beng Chin Ooi, and Kian-Lee Tan. Querying high-dimensional data in single-dimensional space. *The VLDB Journal*, 13(2):105–119, 2004.

[YH06] Jun Yang and Alexander G. Hauptmann. Annotating News Video with Locations. In *CIVR*, pages 153–162, 2006.

[YH07] Rong Yan and Alexander G. Hauptmann. A review of text and image retrieval approaches for broadcast news video. *Inf. Retr.*, 10(4-5):445–484, 2007.

[YI99] Atsuo Yoshitaka and Tadao Ichikawa. A Survey on Content-Based Retrieval for Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, 1999.

[YLL$^+$05] Yidong Yuan, Xuemin Lin, Qing Liu, Wei Wang, Jeffrey Xu Yu, and Qing Zhang. Efficient computation of the skyline cube. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 241–252. VLDB Endowment, 2005.

[YSLH03] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003. ACM.

[ZC03] C. Zhang and T. Chen. *From Low Level Features to High Level Semantics*, chapter 27. CRC Press, 2003.

[ZG02] Rong Zhao and William I. Grosky. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Transactions on Multimedia*, 4(2):189–200, 2002.

[ZL01] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, New York, NY, USA, 2001. ACM.

Appendix A

Audio descriptors

```
<feature>
<name>Spectral Centroid Overall Standard Deviation</name>
<v>8.81E-1</v>
</feature>
<feature>
<name>Spectral Rolloff Point Overall Standard Deviation</name>
<v>1.137E-2</v>
</feature>
<feature>
<name>Spectral Flux Overall Standard Deviation</name>
<v>1.363E-4</v>
</feature>
<feature>
<name>Compactness Overall Standard Deviation</name>
<v>3.742E1</v>
</feature>
<feature>
<name>Spectral Variability Overall Standard Deviation</name>
<v>1.328E-4</v>
</feature>
<feature>
<name>Root Mean Square Overall Standard Deviation</name>
<v>4.869E-3</v>
</feature>
<feature>
<name>Fraction Of Low Energy Windows Overall Standard Deviation</name>
<v>0E0</v>
</feature>
<feature>
<name>Zero Crossings Overall Standard Deviation</name>
<v>2.354E0</v>
</feature>
```

```
<feature>
<name>Strongest Beat Overall Standard Deviation</name>
<v>0E0</v>
</feature>
<feature>
<name>Beat Sum Overall Standard Deviation</name>
<v>0E0</v>
</feature>
<feature>
<name>Strength Of Strongest Beat Overall Standard Deviation</name>
<v>0E0</v>
</feature>
<feature>
<name>MFCC Overall Standard Deviation</name>
<v>1.538E0</v>
<v>2.536E-1</v>
<v>2.873E-1</v>
<v>4.181E-1</v>
<v>2.302E-1</v>
<v>2.241E-1</v>
<v>2.108E-1</v>
<v>2.2E-1</v>
<v>1.853E-1</v>
<v>1.798E-1</v>
<v>1.889E-1</v>
<v>3.092E-1</v>
<v>1.908E-1</v>
</feature>
<feature>
<name>LPC Overall Standard Deviation</name>
<v>5.342E-3</v>
<v>7.414E-3</v>
<v>7.732E-3</v>
<v>1.35E-2</v>
<v>2.414E-2</v>
<v>2.129E-2</v>
<v>2.484E-2</v>
<v>1.561E-2</v>
<v>1.387E-2</v>
<v>0E0</v>
</feature>
<feature>
<name>Method of Moments Overall Standard Deviation</name>
<v>1.664E-2</v>
<v>3.046E-3</v>
<v>7.762E-1</v>
```

```
<v>1.977E2</v>
<v>5.037E4</v>
</feature>
<feature>
<name>Spectral Centroid Overall Average</name>
<v>2.988E1</v>
</feature>
<feature>
<name>Spectral Rolloff Point Overall Average</name>
<v>2.702E-1</v>
</feature>
<feature>
<name>Spectral Flux Overall Average</name>
<v>1.112E-3</v>
</feature>
<feature>
<name>Compactness Overall Average</name>
<v>1.614E3</v>
</feature>
<feature>
<name>Spectral Variability Overall Average</name>
<v>3.064E-3</v>
</feature>
<feature>
<name>Root Mean Square Overall Average</name>
<v>1.293E-1</v>
</feature>
<feature>
<name>Fraction Of Low Energy Windows Overall Average</name>
<v>0E0</v>
</feature>
<feature>
<name>Zero Crossings Overall Average</name>
<v>7.715E1</v>
</feature>
<feature>
<name>Strongest Beat Overall Average</name>
<v>0E0</v>
</feature>
<feature>
<name>Beat Sum Overall Average</name>
<v>0E0</v>
</feature>
<feature>
<name>Strength Of Strongest Beat Overall Average</name>
<v>0E0</v>
```

```
</feature>
<feature>
<name>MFCC Overall Average</name>
<v>-9.665E1</v>
<v>1.837E-2</v>
<v>4.196E0</v>
<v>3.599E-1</v>
<v>4.397E-1</v>
<v>8.297E-1</v>
<v>-7.187E-1</v>
<v>4.434E-1</v>
<v>-1.094E0</v>
<v>-3.221E-1</v>
<v>2.238E-6</v>
<v>3.451E-1</v>
<v>-1.33E-1</v>
</feature>
<feature>
<name>LPC Overall Average</name>
<v>-8.624E-1</v>
<v>4.964E-1</v>
<v>-6.798E-1</v>
<v>4.896E-1</v>
<v>-3.648E-1</v>
<v>2.661E-1</v>
<v>2.201E-2</v>
<v>-1.149E-1</v>
<v>1.266E-1</v>
<v>0E0</v>
</feature>
<feature>
<name>Method of Moments Overall Average</name>
<v>4.129E-1</v>
<v>1.536E-2</v>
<v>3.916E0</v>
<v>9.981E2</v>
<v>2.545E5</v>
</feature>
```

Appendix B

The metric axioms

Definition: A metric on a set X is a function $d : X \times X \rightarrow R$ (where R is the set of real numbers), which is required to satisfy the following conditions for all A, B and C in X:

- $d(A,A) = d(B,B) = 0$ Constancy of Self-Similarity
- $d(A,B) > d(A,A) = 0$, if A <> B Minimality (Positivity)
- $d(A,B) = d(B,A)$ Symmetry
- $d(A,B) \leq d(A,C) + d(B,C)$ Triangle Inequality

The $d(A,B)$ is also called the distance or dissimilarity between A and B.

Kendall tau distance

From the set of N permutations of λ, let σ and τ be two of them. The *Kendall tau* distance $K(\sigma, \tau)$ between σ and τ is defined to be the number of item pairs (x,y) such that either $\sigma(x) > \sigma(y)$ but $\tau(x) < \tau(y)$ or $\sigma(x) < \sigma(y)$ but $\tau(x) > \tau(y)$. The *Kendall tau* distance can be also seen as the number of pair-wise adjacent transpositions needed to transform one list into the other.

The edit metric

The edit distance between two strings S and T is defined as the minimum number of delete, insert, and substitute operations needed to transform S into T.

Appendix C

LSCOM-Lite concepts

1. Sports: Shots depicting any sport in action
2. Entertainment
3. Weather: Shots depicting any weather related news or bulletin
4. Court: Shots of the interior of a court-room location
5. Office: Shots of the interior of an office setting
6. Meeting: Shots of a Meeting taking place indoors
7. Studio: Shots of the studio setting including anchors, interviews and all events that happen in a news room
8. Outdoor: Shots of Outdoor locations
9. Building: Shots of an exterior of a building
10. Desert: Shots with the desert in the background
11. Vegetation: Shots depicting natural or artificial greenery, vegetation woods, etc.
12. Mountain: Shots depicting a mountain or mountain range with the slopes visible
13. Road: Shots depicting a road
14. Sky: Shots depicting sky
15. Snow: Shots depicting snow
16. Urban: Shots depicting an urban or suburban setting
17. Waterscape_Waterfront: Shots depicting a waterscape or waterfront
18. Crowd: Shots depicting a crowd

19. Face: Shots depicting a face
20. Person: Shots depicting a person (the face may or may not be visible)
21. Government-Leader
22. Corporate-Leader
23. Police_Security: Shots depicting law enforcement or private security agency personnel
24. Military: Shots depicting the military personnel
25. Prisoner: Shots depicting a captive person, e.g., imprisoned, behind bars, in jail or in handcuffs, etc.
26. Animal: Shots depicting an animal, not counting a human as an animal
27. Computer_TV-screen:Shots depicting a television or computer screen
28. Flag-US: Shots depicting a US flag
29. Airplane: Shots of an airplane
30. Car: Shots of a car
31. Bus: Shots of a bus
32. Truck: Shots of a truck
33. Boat_Ship: Shots of a boat or ship
34. Walking_Running: Shots depicting a person walking or running
35. People-Marching: Shots depicting many people marching as in a parade or a protest
36. Explosion_Fire: Shots of an explosion or a fire
37. Natural-Disaster: Shots depicting the happening or aftermath of a natural disaster such as earthquake, flood, hurricane, tornado, tsunami
38. Maps: Shots depicting regional territory graphically as a geographical or political map
39. Charts: Shots depicting any graphics that is artificially generated such as bar graphs, line charts, etc. (maps should not be included)

Appendix D

The TRECVID topics

There have been 24 TRECVID topics expressing various information needs. Numbered from 197 to 220, the topics are bundles of text formulations, image and video examples. The table below contains only the text parts of the topics.

Table D.1: The TRECVID topics

Topic number	Text Description
197	“Find shots of one or more people walking up stairs”
198	“Find shots of a door being opened”
199	“Find shots of a person walking or riding a bicycle”
200	“Find shots of hands at a keyboard typing or using a mouse”
201	“Find shots of a canal, river, or stream with some of both banks visible”
202	“Find shots of a person talking on a telephone”
203	“Find shots of a street market scene”
204	“Find shots of a street protest or parade”
205	“Find shots of a train in motion”
206	“Find shots with hills or mountains visible”
207	“Find shots of waterfront with water and buildings”
208	“Find shots of a street at night”
209	“Find shots with 3 or more people sitting at a table”
210	“Find shots with one or more people walking with one or more dogs”
211	“Find shots with sheep or goats”
212	“Find shots in which a boat moves past”
213	“Find shots of a woman talking toward the camera in an interview - no other people visible”

Continued on Next Page…

Table D.1 – Continued

Topic number	Text Description
214	“Find shots of a very large crowd of people (fills more than half of field of view)”
215	“Find shots of a classroom scene with one or more students”
216	“Find shots of a bridge”
217	“Find shots of a road taken from a moving vehicle through the front windshield”
218	“Find shots of one or more people playing musical instruments such as drums, guitar, flute, keyboard, piano, etc.”
219	“Find shots that contain the Cook character in the Klokhuis series”
220	“Find grayscale shots of a street with one or more buildings and one or more people”

Appendix E

Obtaining the syntactic structure of an NLP query

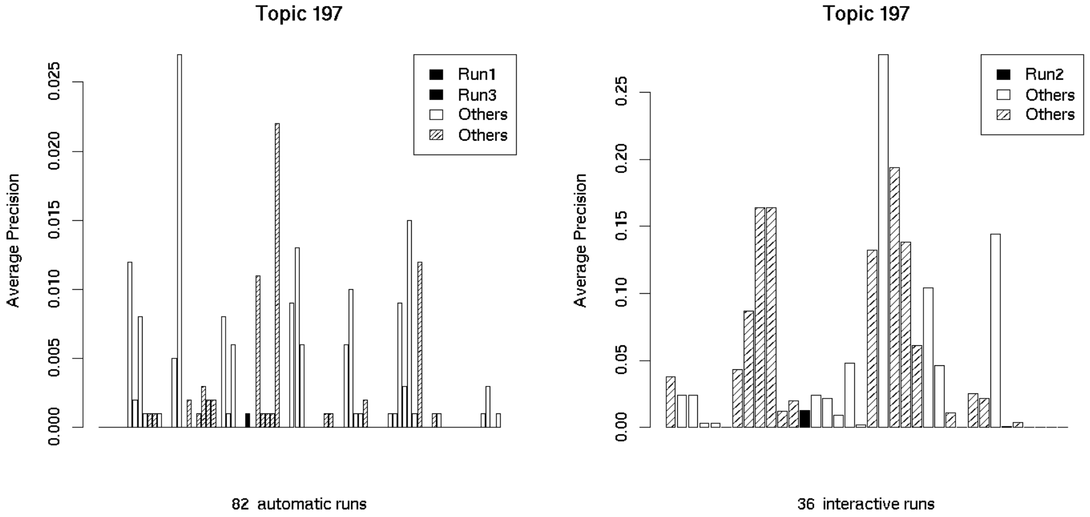
An output example of the VISL (Visual Interactive Syntax Learning) parser for a given natural language input, such as “Find shots with one or more emergency vehicles in motion (e.g., ambulance, police car, fire truck, etc.)”, is presented below.

```
coord(’0173’,1,1,w(’Find’,’find’,’<mv>’,’V’,’IMP’,’@FS-COM’,
’#1->0’)).
coord(’0173’,1,2,w(’shots’,’shot’,’N’,’P’,’NOM’,’@<ACC’,’#2->1’)).
coord(’0173’,1,3,w(’with’,’with’,’PRP’,’@<ADVL’,’#3->1’)).
coord(’0173’,1,4,w(’one=or=more’,’[one=or=more]’,
’ADJ’,’POS’,’@>N’,’#4->8’)).
coord(’0173’,1,5,w(’emergency’,’emergency’,’N’,’S’,’NOM’,
’@>N’,’#5->8’)).
coord(’0173’,1,6,w(’vehicles’,’vehicle’,’N’,’P’,’NOM’,’@>A’,
’#6->7’)).
coord(’0173’,1,7,w(’in’,’in’,’ADJ’,’POS’,’@>N’,’#7->8’)).
coord(’0173’,1,8,w(’motion’,’motion’,’N’,’S’,’NOM’,
’@P<’,’#8->3’)).
coord(’0173’,1,9,w(’e.g.’,’[e.g.]’,’ADV’,
’@ADVL>’,’#10->17’)).
coord(’0173’,1,10,w(’ambulance’,’ambulance’,’N’,’S’,’NOM’,
’@SUBJ>’,’#12->17’)).
coord(’0173’,1,11,w(’police’,’police’,’N’,’S’,’NOM’,
’@>N’,’#14->15’)).
coord(’0173’,1,12,w(’car’,’car’,’N’,’S’,’NOM’,’@SUBJ>’,
’#15->12’)).
coord(’0173’,1,13,w(’fire’,’fire’,’<mv>’,’V’,’IMP’,’@FS-<ADVL’,’fire’,
’<mv>’,  ’V’,’PR’,’-3S’,’@FS-<ADVL’,
’#17->1’)).
coord(’0173’,1,14,w(’truck’,’truck’,’N’,’S’,’NOM’,
’@<ACC’,’#18->17’)).
```

Appendix F

The TRECVID results per topic

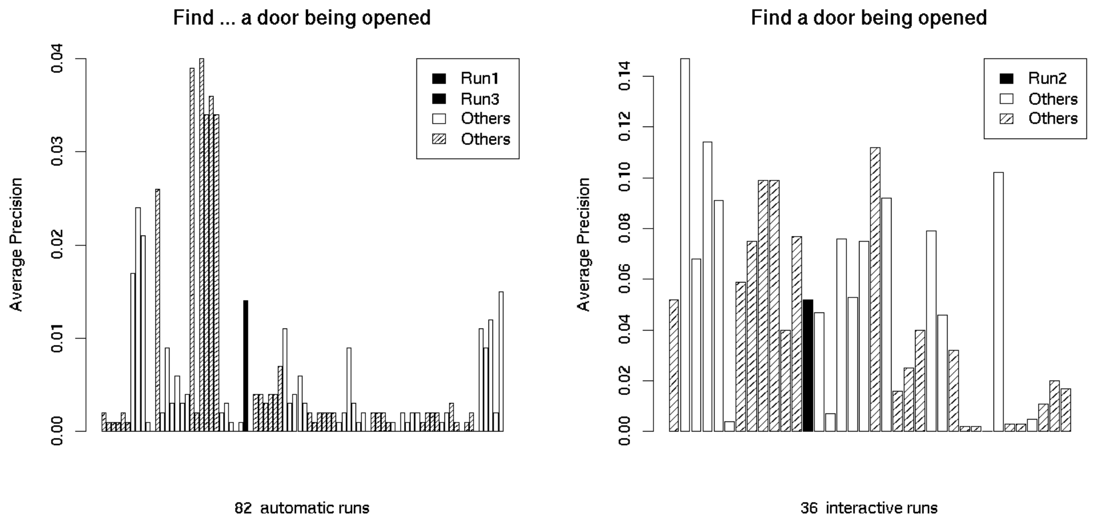
Each of the following pairs of figures correspond to a different TRECVID topic. The figure pairs are presented in the topic order. They present in parallel the AP scores of the automatic and the interactive runs.



(a) Automatic

(b) Interactive

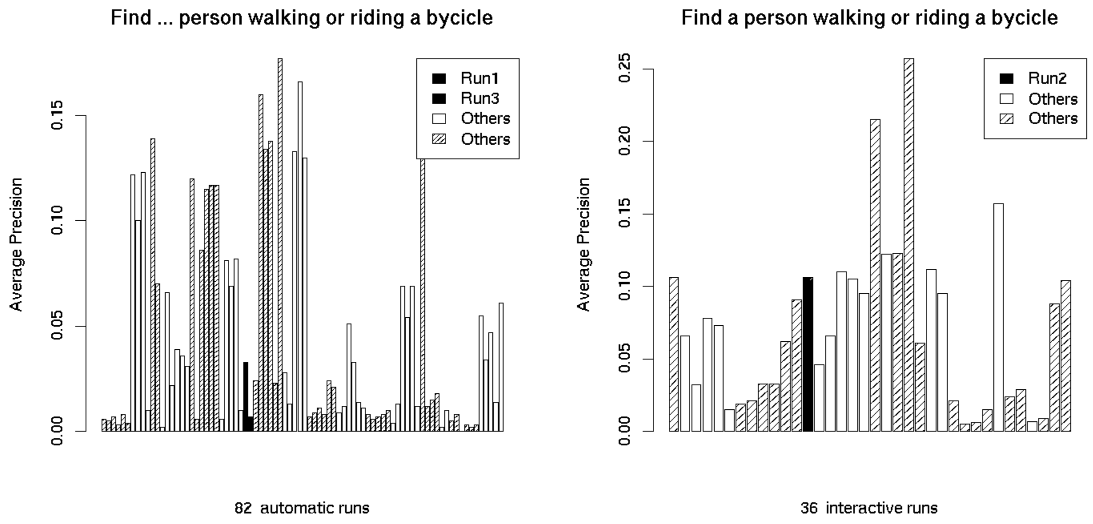
Figure F.1: Topic 197



(a) Automatic

(b) Interactive

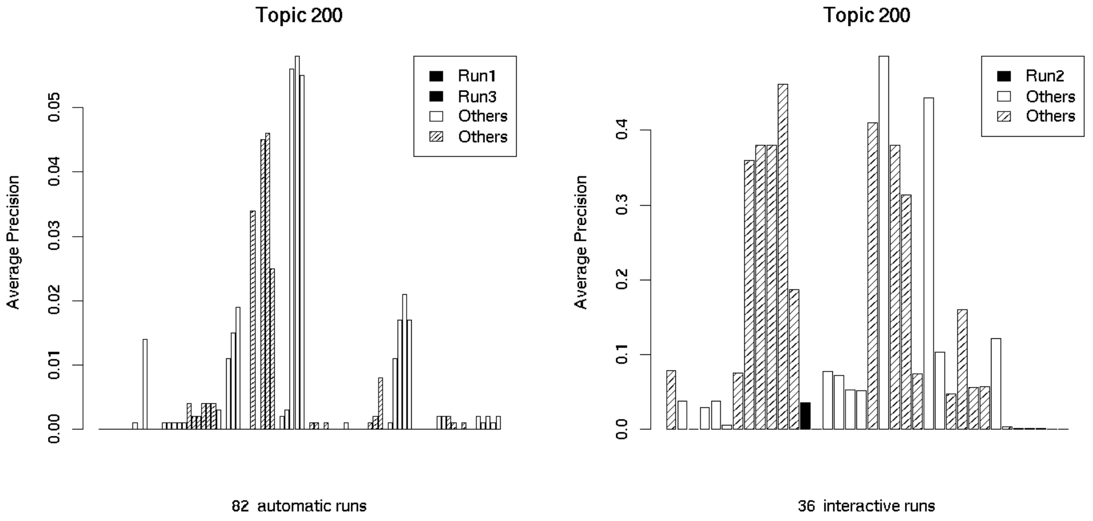
Figure F.2: Topic 198



(a) Automatic

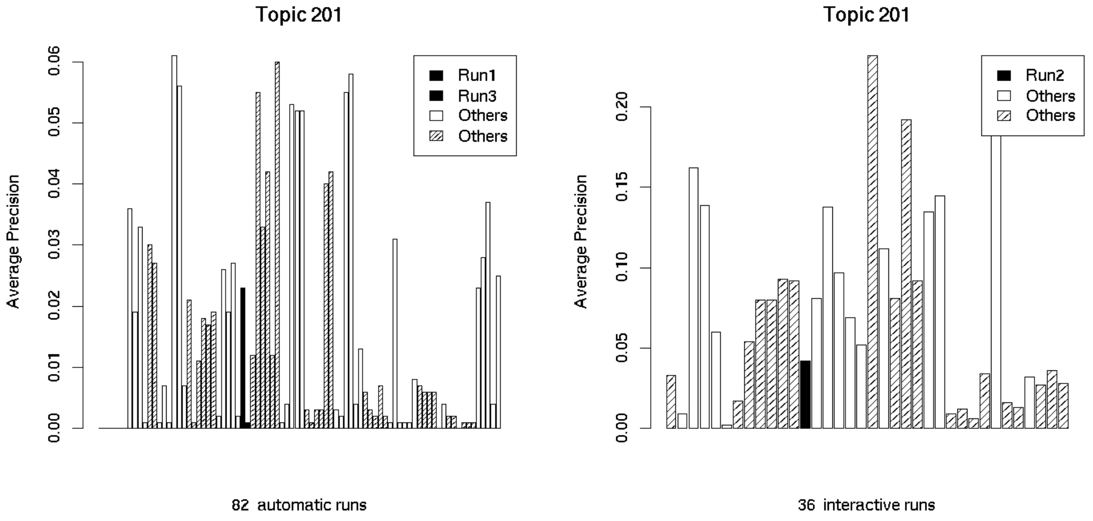
(b) Interactive

Figure F.3: Topic 199



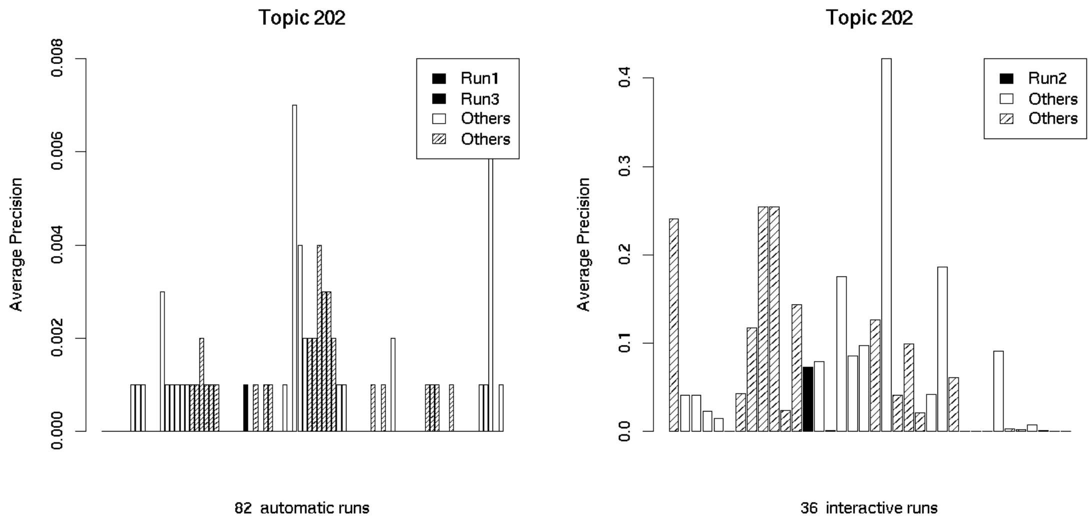
(a) Automatic (b) Interactive

Figure F.4: Topic 200



(a) Automatic (b) Interactive

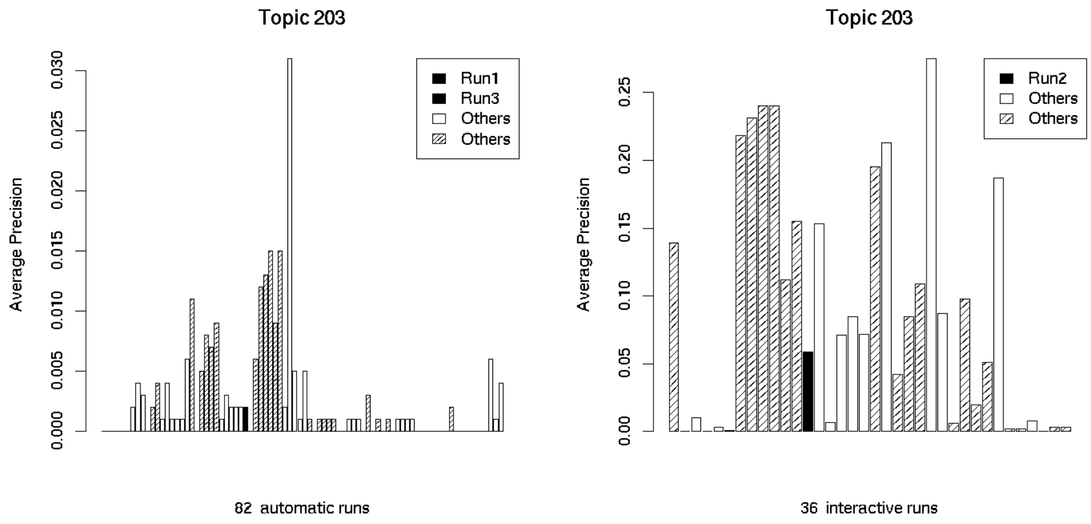
Figure F.5: Topic 201



(a) Automatic

(b) Interactive

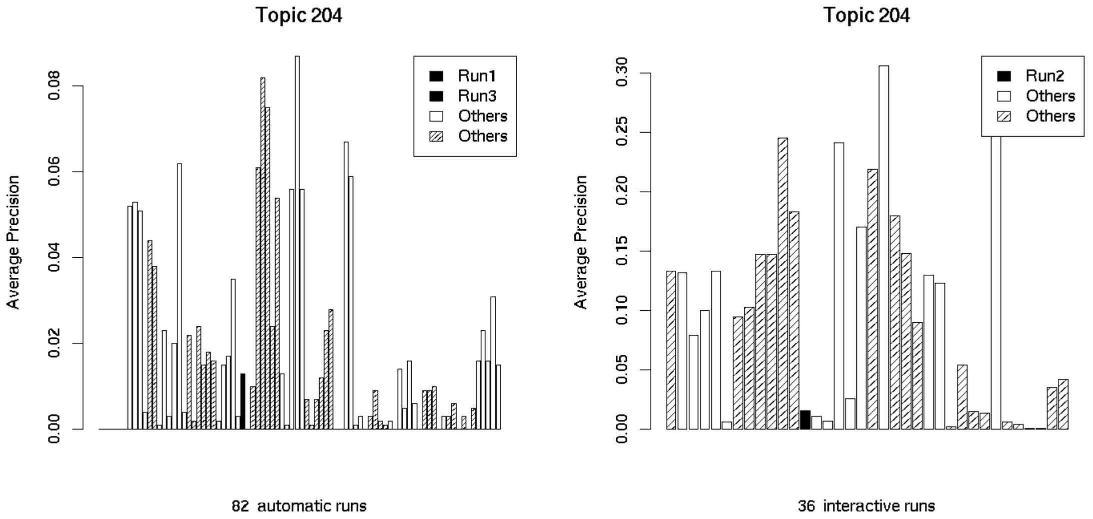
Figure F.6: Topic 202



(a) Automatic

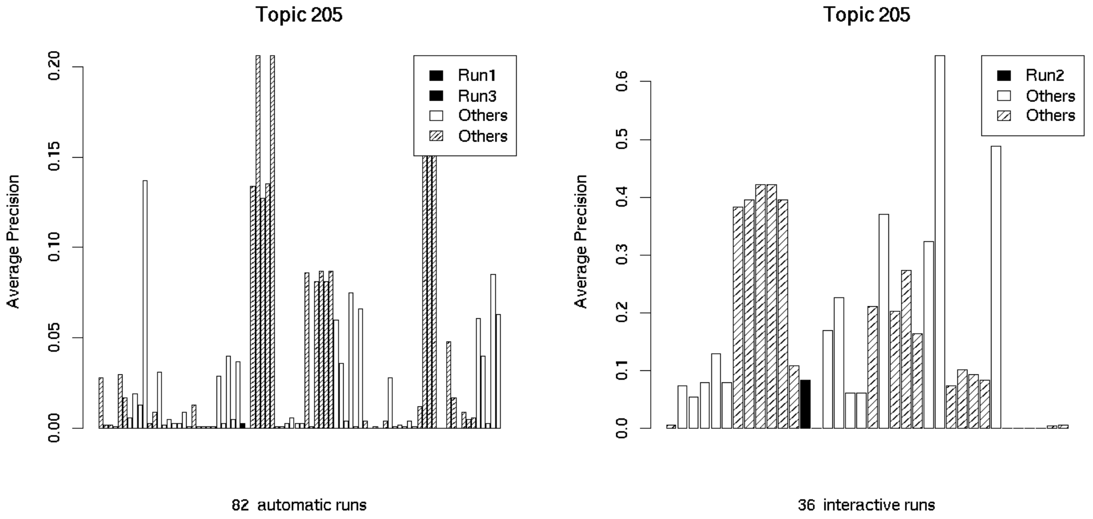
(b) Interactive

Figure F.7: Topic 203



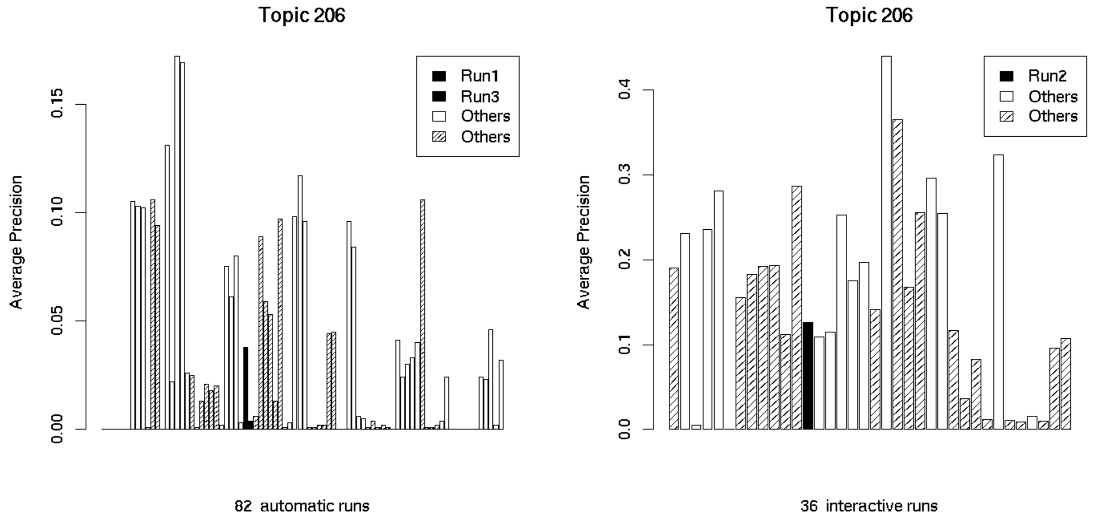
(a) Automatic (b) Interactive

Figure F.8: Topic 204



(a) Automatic (b) Interactive

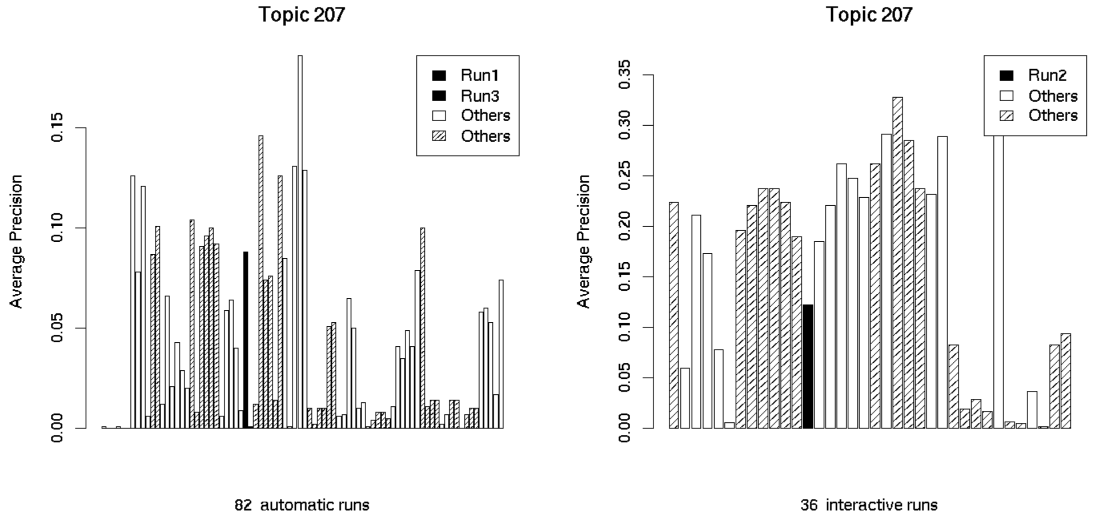
Figure F.9: Topic 205



(a) Automatic

(b) Interactive

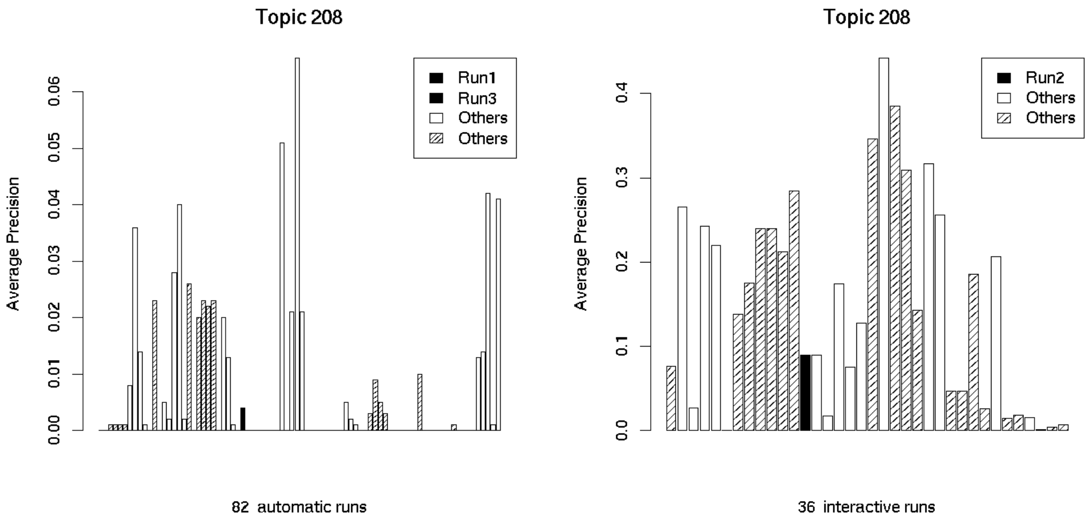
Figure F.10: Topic 206



(a) Automatic

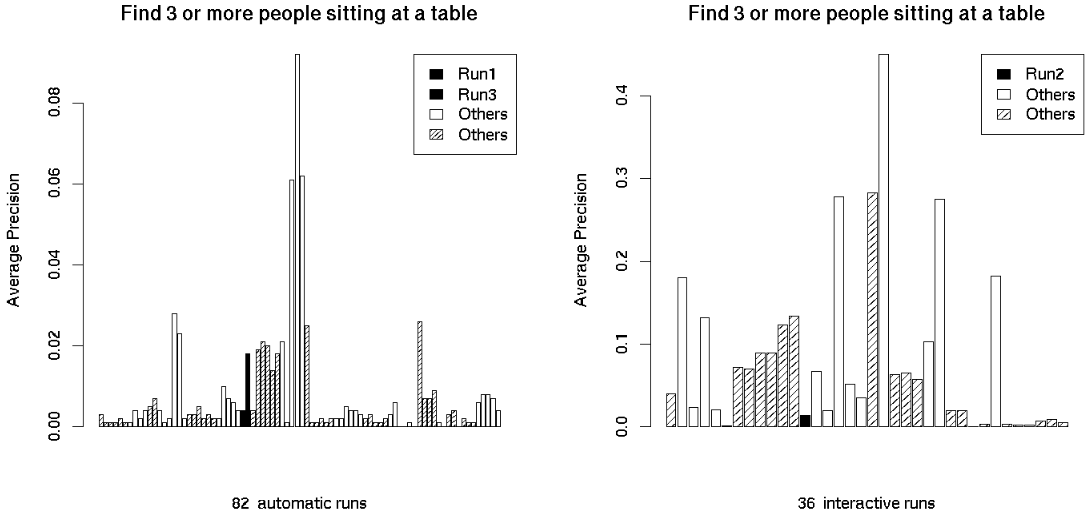
(b) Interactive

Figure F.11: Topic 207



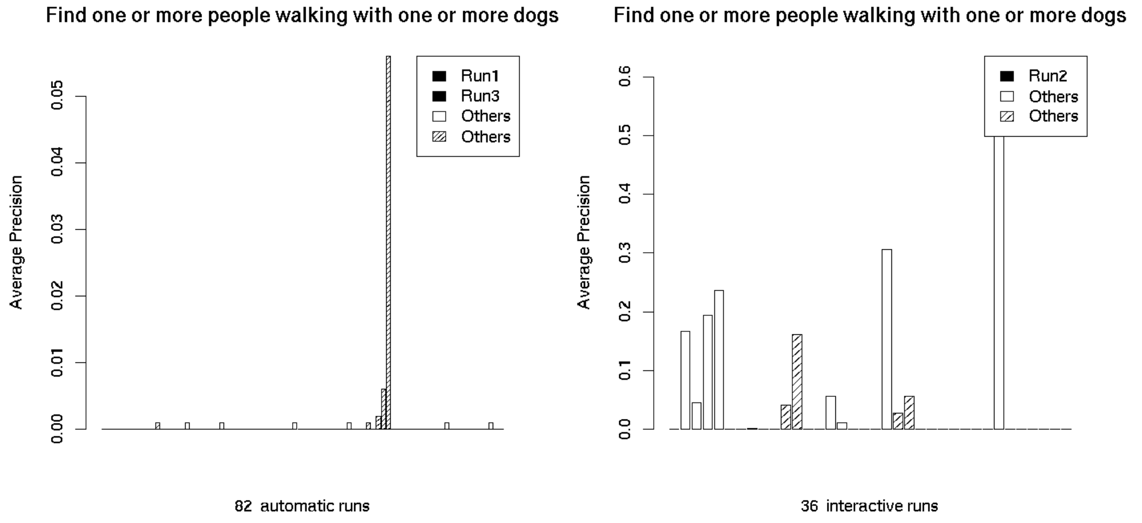
(a) Automatic (b) Interactive

Figure F.12: Topic 208



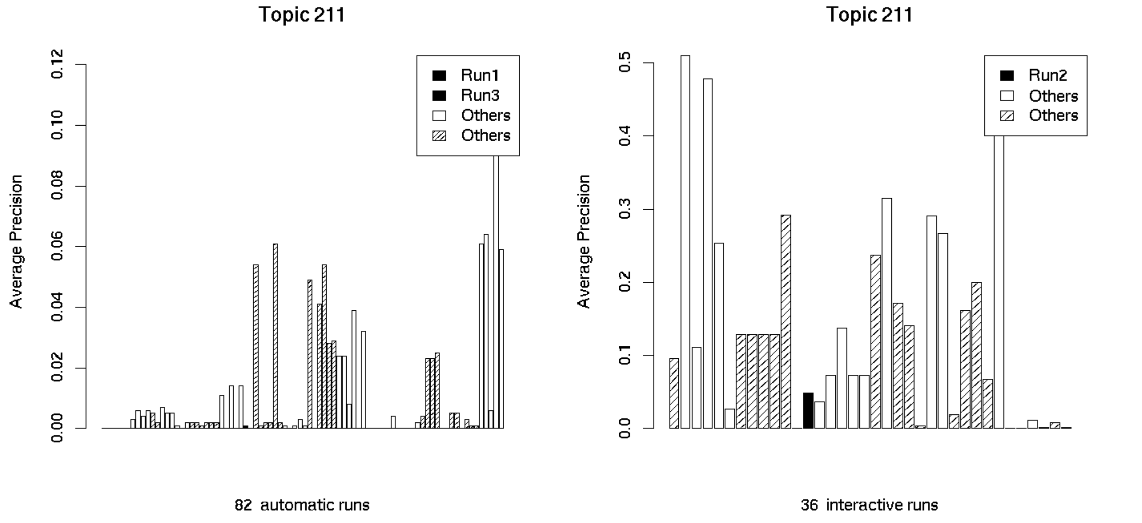
(a) Automatic (b) Interactive

Figure F.13: Topic 209



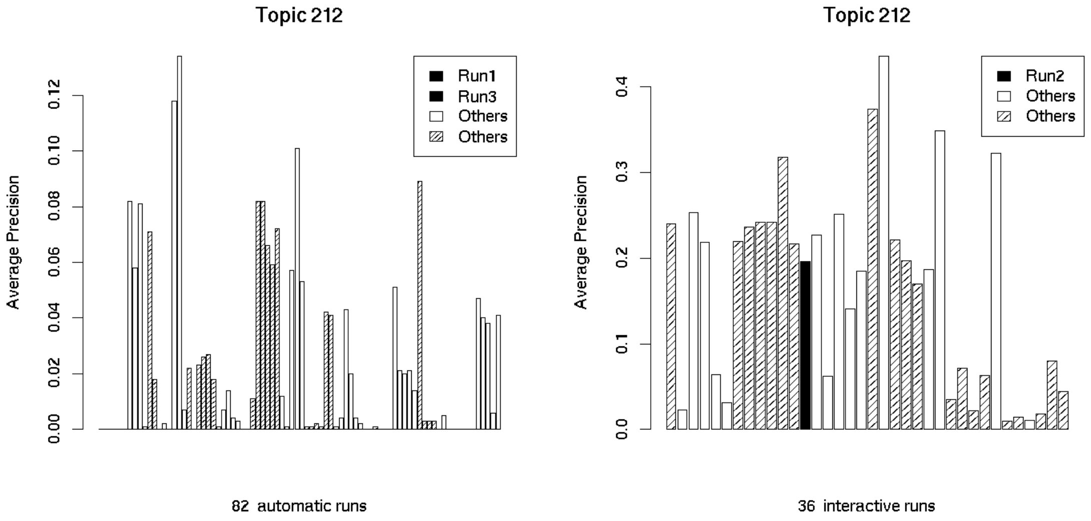
(a) Automatic (b) Interactive

Figure F.14: Topic 210



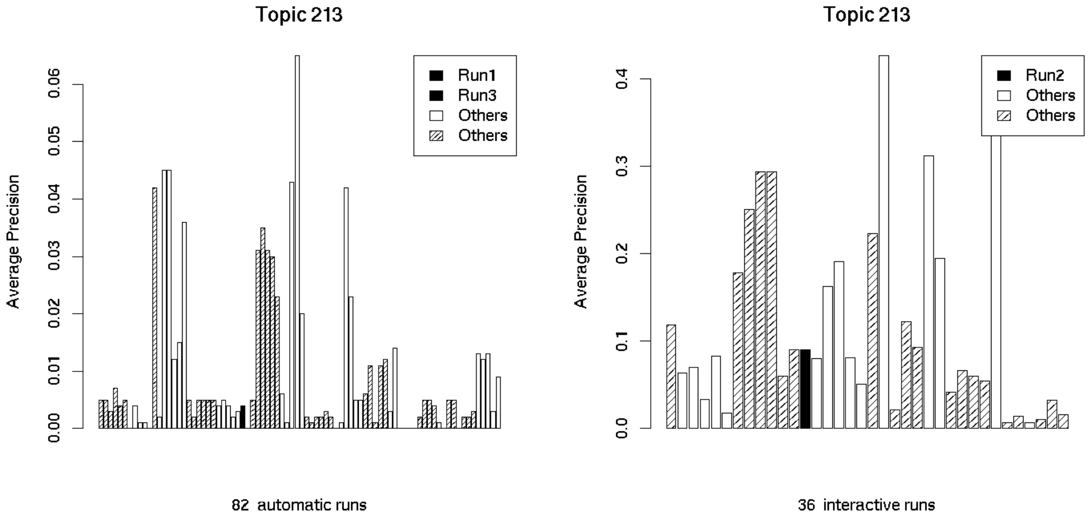
(a) Automatic (b) Interactive

Figure F.15: Topic 211



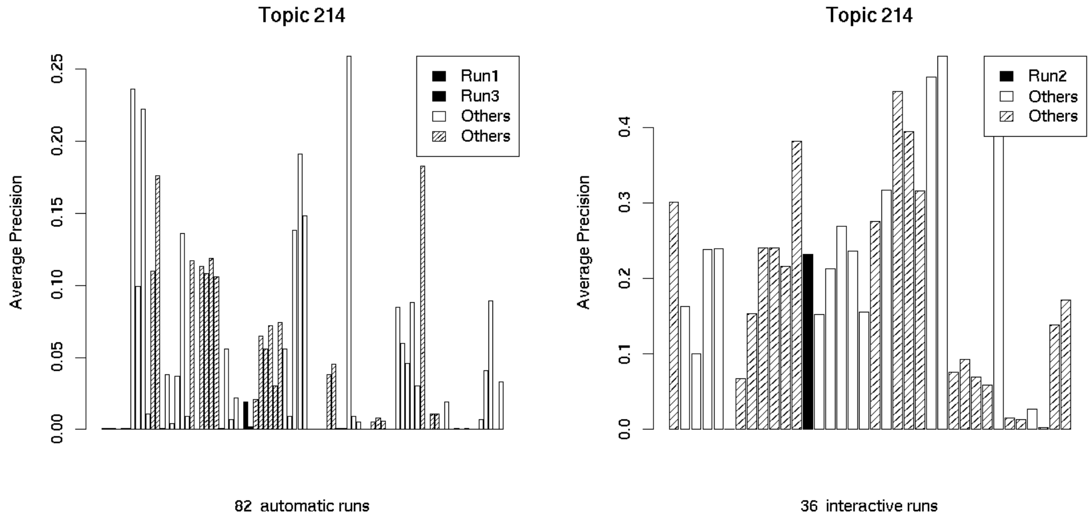
(a) Automatic (b) Interactive

Figure F.16: Topic 212



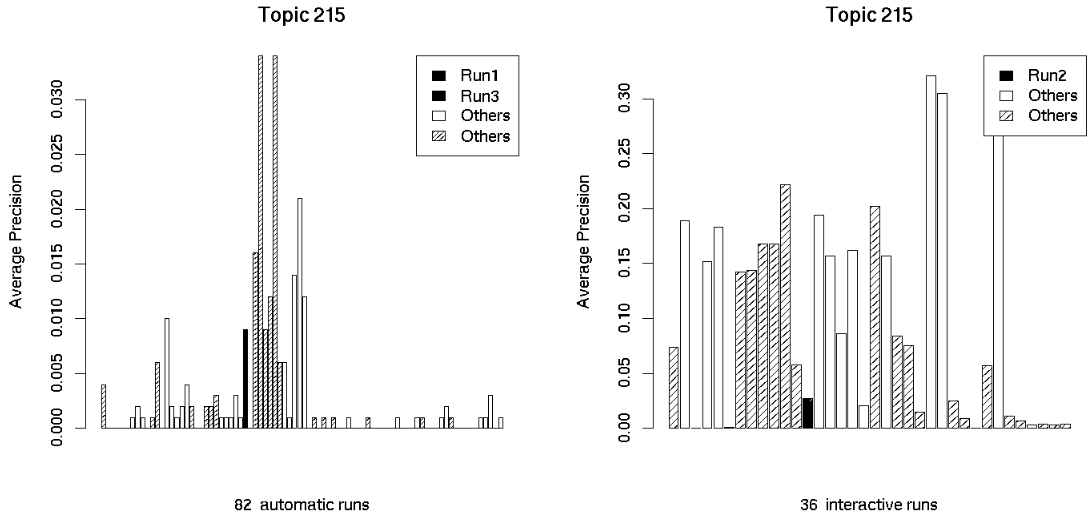
(a) Automatic (b) Interactive

Figure F.17: Topic 213



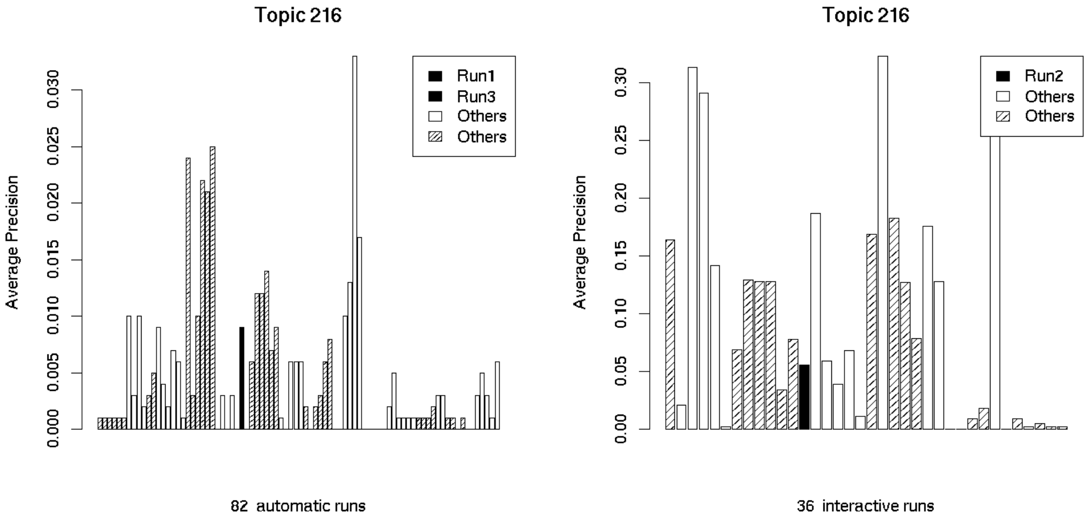
(a) Automatic (b) Interactive

Figure F.18: Topic 214



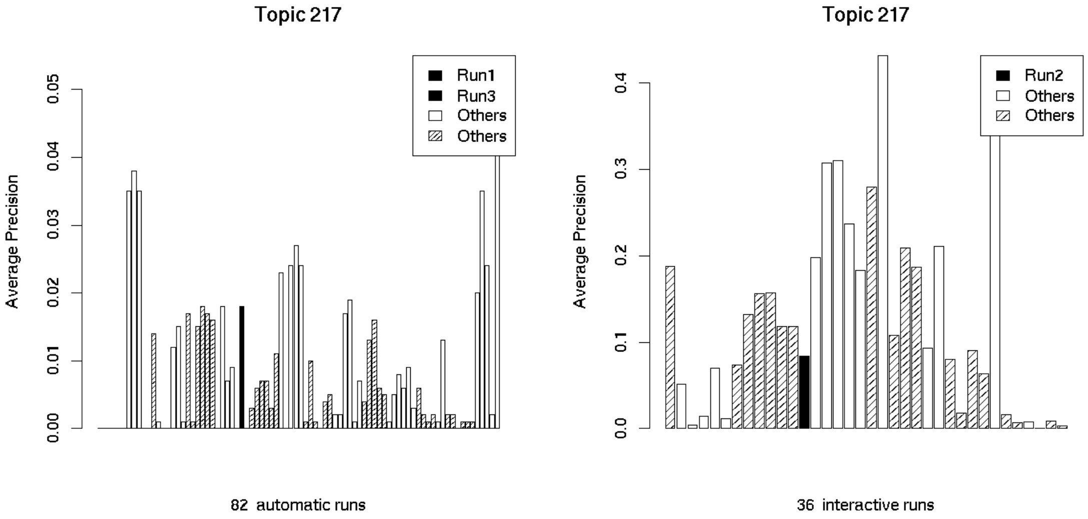
(a) Automatic (b) Interactive

Figure F.19: Topic 215



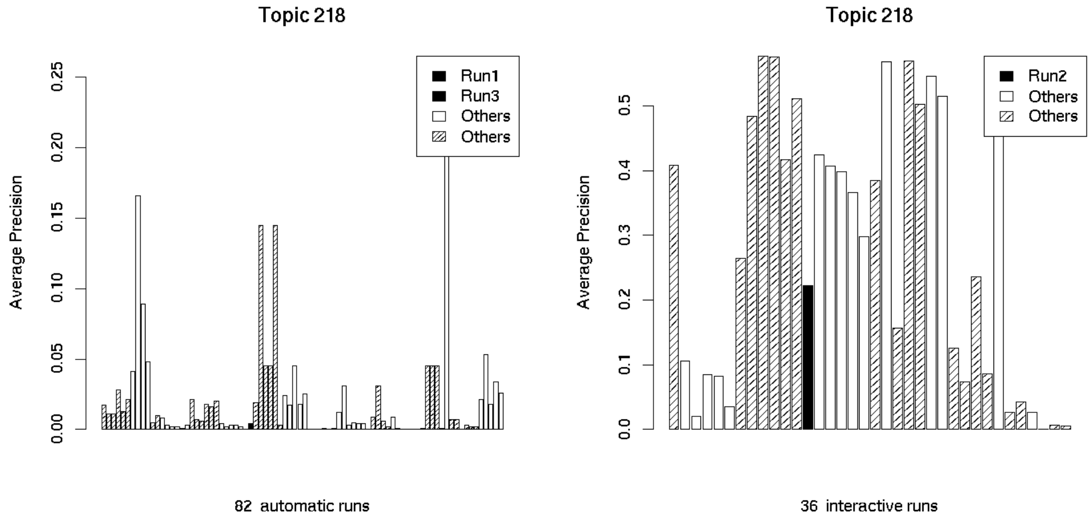
(a) Automatic (b) Interactive

Figure F.20: Topic 216



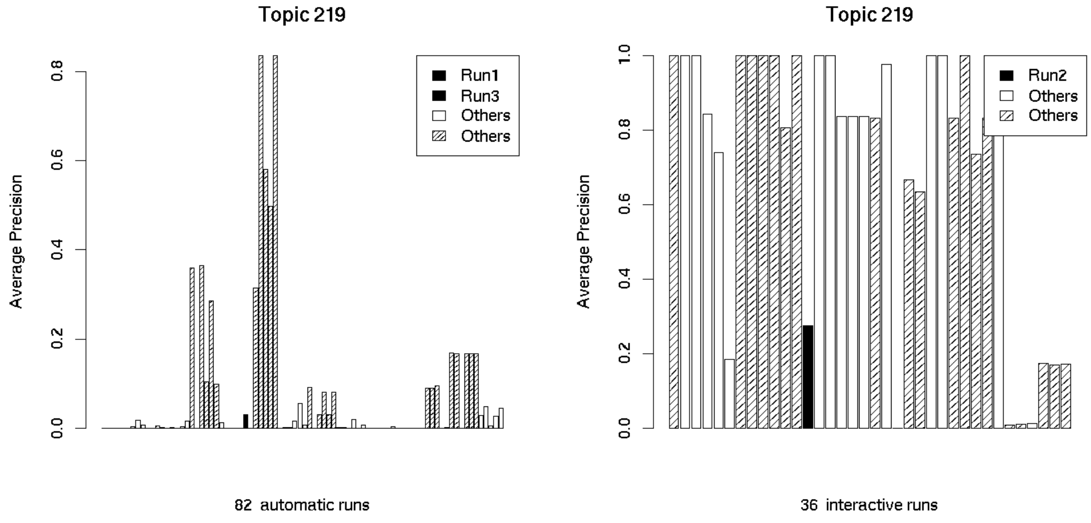
(a) Automatic (b) Interactive

Figure F.21: Topic 217



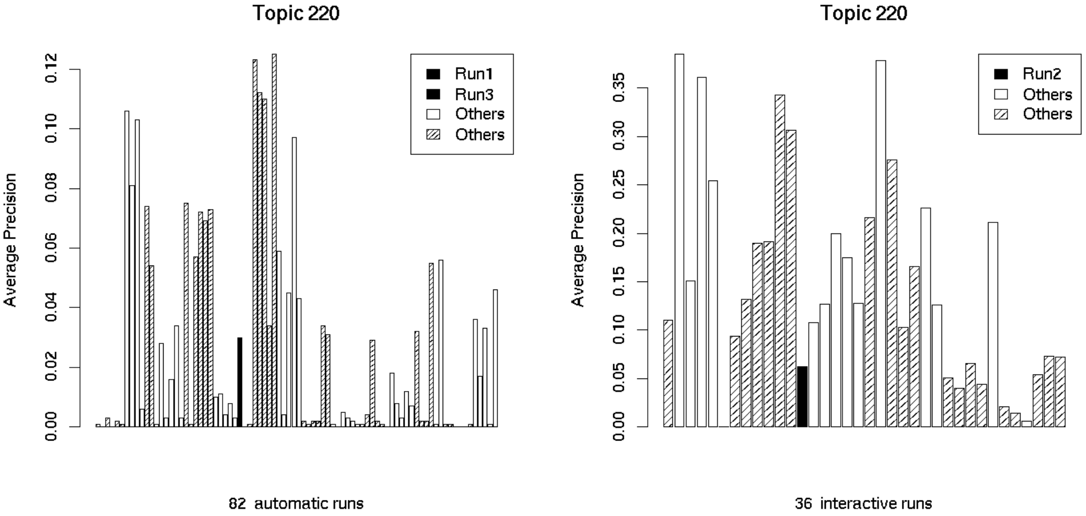
(a) Automatic (b) Interactive

Figure F.22: Topic 218



(a) Automatic (b) Interactive

Figure F.23: Topic 219



(a) Automatic

(b) Interactive

Figure F.24: Topic 220

Appendix G

A sample retrieval session

A video is available on CD.

Index