University of Passau
Department of Informatics and Mathematics
Chair of Distributed Information Systems

Doctoral Thesis

Unified Retrieval in Distributed and Heterogeneous Multimedia Information Systems

Dipl. Inf. Florian Stegmaier

December 18, 2013

Advisor: Prof. Dr. Harald Kosch
Second Advisor: Prof. Dr. Richard Chbeir

In theory, there is no difference between theory and practice.
But, in practice, there is.
Jan L. A. van de Snepscheut (1953 – 1994).

For Tamara.

Abstract

Multimedia retrieval is an essential part of today's world. This situation is observable in industrial domains, e.g., medical imaging, as well as in the private sector, visible by activities in manifold Social Media platforms. This trend led to the creation of a huge environment of multimedia information retrieval services offering multimedia resources for almost any user requests. Indeed, the encompassed data is in general retrievable by (proprietary) APIs and query languages, but unfortunately a unified access is not given due to arising interoperability issues between those services. In this regard, this thesis focuses on two application scenarios, namely a medical retrieval system supporting a radiologist's workflow, as well as an interoperable image retrieval service interconnecting diverse data silos.

The scientific contribution of this dissertation is split in three different parts: the first part of this thesis improves the metadata interoperability issue. Here, major contributions to a community-driven, international standardization have been proposed leading to the specification of an API and ontology to enable a unified annotation and retrieval of media resources. The second part issues a metasearch engine especially designed for unified retrieval in distributed and heterogeneous multimedia retrieval environments. This metasearch engine is capable of being operated in a federated as well as autonomous manner inside the aforementioned application scenarios. The remaining third part ensures an efficient retrieval due to the integration of optimization techniques for multimedia retrieval in the overall query execution process of the metasearch engine.

Keywords: Distributed multimedia retrieval, query optimization, multimedia annotation, interoperability

Kurzzusammenfassung

Egal ob im industriellen Bereich oder auch im Social Media - multimediale Daten nehmen eine immer zentralere Rolle ein. Aus diesem fortlaufendem Entwicklungsprozess entwickelten sich umfangreiche Informationssysteme, die Daten für zahlreiche Bedürfnisse anbieten. Allerdings ist ein einheitlicher Zugriff auf jene verteilte und heterogene Landschaft von Informationssystemen in der Praxis nicht gewährleistet. Und dies, obwohl die Datenbestände meist über Schnittstellen abrufbar sind. Im Detail widmet sich diese Arbeit mit der Bearbeitung zweier Anwendungsszenarien. Erstens, einem medizinischen System zur Diagnoseunterstützung und zweitens einer interoperablen, verteilten Bildersuche.

Der wissenschaftliche Teil der vorliegenden Dissertation gliedert sich in drei Teile: Teil eins befasst sich mit dem Problem der Interoperabilität zwischen verschiedenen Metadatenformaten. In diesem Bereich wurden maßgebliche Beiträge für ein internationales Standardisierungsverfahren entwickelt. Ziel war es, einer Ontologie, sowie einer Programmierschnittstelle einen vereinheitlichten Zugriff auf multimediale Informationen zu ermöglichen. In Teil zwei wird eine externe Metasuchmaschine vorgestellt, die eine einheitliche Anfrageverarbeitung in heterogenen und verteilten Multimediadatenbanken ermöglicht. In den Anwendungsszenarien wird zum einen auf eine föderative, als auch autonome Anfrageverarbeitung eingegangen. Abschließend werden in Teil drei Techniken zur Optimierung von verteilten multimedialen Anfragen präsentiert.

Stichwörter: Verteilte Multimedia-Anfrageverarbeitung, Optimierung von Multimedia-Anfragen, Multimedia Annotation, Interoperabilität

Acknowledgements

I have been working on the topics of this thesis for more then four years. Without the constant support of certain people, I surely would not have had the patience to complete it. I am not a man of many words or big explanations; however, I want to take the opportunity to express my grateful thanks:

At first I want to thank my advisor Harald Kosch as well as Mario Döller for their supervision, sophisticated feedback and friendly advise. Honestly, I feel very lucky to have met so many outstanding colleagues during my period at the Chair of Distributed Information Systems (in alphabetic order): Werner Bailer, Sebastian Bayerl, Emanuel Berndl, Tobias Bürger, David Coquil, Andreas Eisenkolb, Michael Granitzer, Udo Gröbner, Thomas Kurz, Tobias Rene Mayer, Britta Meixner, Tilmann Rabl, Hatem Moussely Sergieh, Kai Schlegel, Stella Stars, Ingrid Winter, and Andreas Wölfl. All of you are a part of this thesis: the feedback you gave me, the ears you lent me during hard times or simply because you always brightened my day. You are all very special to me.

Further, I want to thank my family: My parents Christa and Johann, who always believed in me and gave me the opportunity to study in the area of my interest. My brother Bernhard, who had really hard times in guiding my first steps in programming Java, along with his wife Angela. My aunts and uncles Uschi and Hannes as well as Anita and Raimund. Finally my grandma Paula for being as she is. Besides, I am blessed with really great friends, on which I can always count – unfortunately I cannot enumerate everybody, the ones I have in mind are aware of it.

I dedicate this thesis to my beloved wife Tamara, the most important person in my life. Thank you for everything.

Contents

List of Tables

List of Figures

List of Listings

Part I

Preface

Chapter 1

Introduction

1.1 Motivation

Nowadays, the public interest in multimedia information retrieval is tremendous. This situation can be determined by current statistics of Pingdom[1]: First, anybody has the ability to produce digital multimedia content in an easy fashioned way. Almost every mobile device is already equipped with image sensors for taking still images or (high resolution) movies. Along with this, the acquisition costs, such as for digital cameras, have decreased dramatically. Second, due to the rise of Social Media [EABC+11], uncountable services for easy multimedia content publishing have arisen for nearly every usage domain. Finally, the gap between production and consumption of multimedia data has closed through cheap and nearly everywhere accessible high-speed internet infrastructures (e.g., UMTS[2]). To be more concrete, the Social Media trend has resulted in a vast amount of blogs (152 million) and social networks with millions of user profiles, approximately 175 million accounts on Twitter and 600 million. on Facebook. Following this trend, several billions of user-generated multimedia resources are publicly available on Social Media sharing platforms such as Flickr[3], Picasa[4] or YouTube[5]. While interacting specifically with a single platform, an user will have the impression that each platform on its own offers sophisticated multimedia retrieval abilities. Unfortunately, each system uses its own (proprietary) data description formats or enables data access on the basis of diverse APIs or query languages. Summing up these facts, the user is hindered in getting a global access on the encompassed data because of interoperability issues.

It has to be stated, that interoperability issues have not emerged within the last decade, nor are limited to the multimedia domain at all. Robertson revisits in [Rob03] research efforts that happened back in 1980 with the aim to find a unified retrieval model. By comparing the models, he concluded that *"[...] the two models seemed to address this question in different and apparently incompatible ways [...]"*. This observation can be applied to the current situation of multimedia retrieval environments in the Web as illustrated in Figure 1.1 in an abstract way. Many promising usage scenarios provoked researchers and industries to investigate

[1] http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, last checked December 18, 2013.

[2] Universal Mobile Telecommunications System

[3] http://www.flickr.com/, last checked December 18, 2013.

[4] http://picasa.google.com/, last checked December 18, 2013.

[5] http://www.youtube.com/, last checked December 18, 2013.

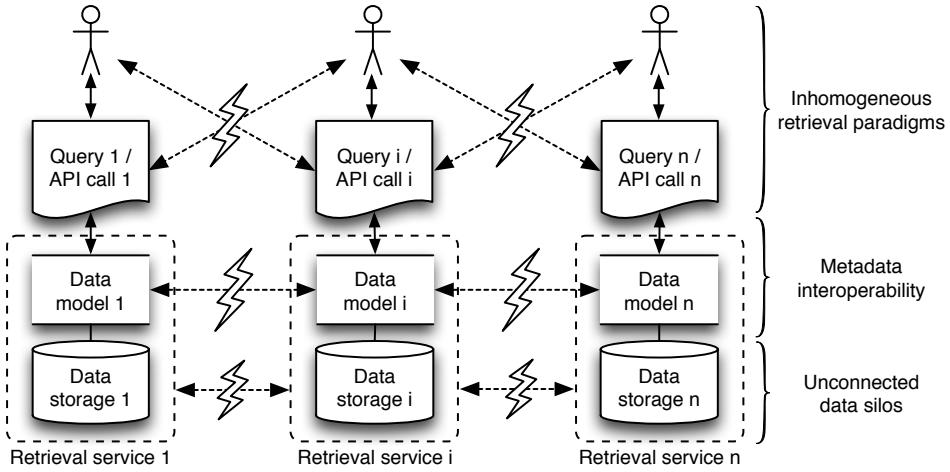
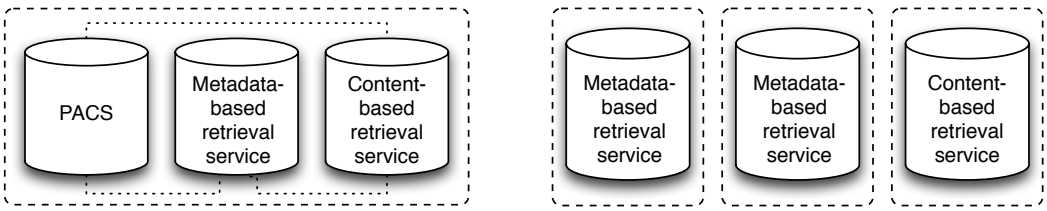


Figure 1.1: Abstract situation of retrieval environments in the Web

solutions and approaches for storing and archiving the immense amount of produced multimedia data - mostly leading to isolated applications operating within very limited domains. The logical prosecution is, that one rapidly ends up in a highly heterogeneous environment of *unconnected data silos*. In series the involved domains feature individual sets of metadata schemes for describing content, technical or structural information of multimedia resources leading to the well-known *metadata interoperability* issue. Furthermore, depending on the management and retrieval requirements, these data sets are accessible in different systems supporting a multiple set of retrieval models and query languages leading building a collection of *inhomogeneous retrieval paradigms*. Client applications, e.g., mobile applications, enable the users to query each specific retrieval service to satisfy their information need. Due to the lack of common data models and unified retrieval techniques, the media resources are locked within these silos prevented from a homogeneous access [Smi08]. By summing up all these obstacles, an easy and efficient access and retrieval across those system borders is a very cumbersome task.

To make the abstract representation of Figure 1.1 more concrete, the three identified interoperability dimensions are applied to two real-world application scenarios as introduced in Figure 1.2:



(i) Federated medical retrieval (ii) Isolated image retrieval

Figure 1.2: Unified multimedia retrieval: Real-world application scenarios

Federated medical retrieval. Figure 1.2 (i) considers a medical application assisting radiologists during the diagnostic process. Here, the present knowledge base is divided in three logically linked retrieval services: a Picture Archiving and Communication System (PACS) stores demographic data of patients, as well as the raw media resources consisting of CT scans. A metadata-based retrieval service is able to query information stored in annotations added to the CT scans, e.g., the region of the body or a lesion. Finally, a content-based retrieval service answers with similar media resources to a given input media resource. In this example the interlinking is done with unique object identifiers. A typical retrieval scenario is the following query:

> *"Give me all lesions, which are similar to a given region in an example CT scan, located inside the liver and the patient is at least of age 65!"*

The query evaluation process takes into account all of the three retrieval services. Since the retrieval services expose logical links between each other, a federated retrieval scenario is present. In terms of interoperability issues, inhomogeneous retrieval paradigms as well as metadata interoperability have to be solved.

Isolated image retrieval. In contrast to the aforementioned application scenario, the isolated image retrieval scenario shown in Figure 1.2 (ii) concentrates on unconnected data silos as well as metadata interoperability. For this task, two metadata-based retrieval services utilizing different metadata schemes are considered in combination with a content-based retrieval service. There are no direct logical links established between the retrieval services preventing a federated query processing in the first place. The possibility to formulate an unified query on the global underlying data set would be highly beneficial for users. An example for this would be the following query:

> *"Give me the first ten images that are similar to http://any.uri/-strawberry.jpg or are annotated with the keyword strawberry!"*

Obviously, the focus in this application scenario is to improve the metadata interoperability issue as well as to enable an overall result fusion of partial result sets without existing knowledge of links between the retrieval services.

Both application scenarios serve as a basis for the remaining thesis to improve the presented interoperability issues.

1.2 Contributions

In this thesis, techniques are proposed to enable a unified retrieval in distributed and heterogeneous multimedia information systems [Ste10]. In detail, the aforementioned real-world application scenarios shall be enabled by softening the three dimensions of interoperability shown in Figure 1.1. The contributions of this thesis are divided in three pillars accordingly:

Multimedia metadata interoperability. The basis for unified retrieval is a homogenous view on distributed and heterogeneous retrieval environments. In this regard, central contributions to a international W3C standardization process have been made resulting in the definition of a pivot metadata schema along with an API to access this information. Both specifications are used as basement to align distributed and diverse multimedia metadata schemas to address metadata information in the medical as well as image retrieval application scenarios.

Unified multimedia retrieval. In distributed retrieval environments such as the real-world application scenario, a crucial task is the interconnection of participating retrieval services while hiding the actual diversity of retrieval paradigms. In this case, the diversity is present by inhomogeneous retrieval paradigms and metadata interoperability. The *Architecture for Interoperable Retrieval (AIR)* exactly tackles this situation. It acts as a mediator between client applications and the connected retrieval services while utilizing novel international standards to enable an abstraction layer for client requests as well as the introduced W3C specification to ensure a global view on the federated data set.

Optimization of multimedia retrieval. The two first pillars serve as a basement for the envisioned unified retrieval. In order to ensure efficiency in the real-world scenarios, AIR has to be equipped with meaningful optimization techniques. In this domain, AIR has improved the medical retrieval application scenario by optimization techniques in terms of federated query execution planning and multimedia caching techniques. In the domain of the isolated image retrieval application scenario, a result fusion technique ensures a qualitative fusion of partial result sets even without interlinked retrieval services.

Prototypical implementations have been developed for each of the three pillars to ensure their validity. This thesis focuses on image retrieval [DJLW08], whereas central concepts are applicable to audio [TWV05] or video retrieval [GN08] without loss of generality.

1.3 Overview

This thesis is composed of six parts. The preface gives an introduction to the subject.

Part II is concerned about the foundations of multimedia information retrieval. This part starts in Chapter 2 with a roundup of multimedia information retrieval in general to get the complete context. The following chapters introduce specific and, for this thesis, relevant topics: Chapter 3 highlights techniques and current best practices for media annotation whereas Chapter 4 copes with indexing and accessing of multimedia resources. Chapter 5 is on multimedia database management systems and concludes this part.

Part III to V describe the contributions of this thesis. Here, Chapter 6 explains the issue of metadata interoperability and gives insights into the *Ontology and API for Media Resource 1.0*. A discussion of this contribution along with its appliance in other research projects can be found in Chapter 7. Beside interoperability issues on the modelling level, Chapter 8 is on the *AIR framework* and the way it enables unified retrieval in heterogeneous multimedia environments. The discussion in Chapter 9 indicates the benefits of the framework with respect to the given application scenarios. The last contribution of this thesis is split in two pieces: Chapter 10 deals with the *optimization of query execution* planning in federated as well as autonomous retrieval scenarios inside the AIR framework. In contrast to that, Chapter 11 issues a novel *late multimedia fusion approach* for combining unconnected partial results. A sophisticated evaluation of the AIR framework can be found in Chapter 12.

Part VI closes the thesis with a resumé in Chapter 13.

Part II

Foundations of Multimedia Information Retrieval

Chapter 2

Multimedia Information Retrieval in a Nutshell

Multimedia information retrieval is a very wide research field exposing a variety of research directions, e.g., stemming from signal processing or the information retrieval community. The following chapters are in charge of introducing research directions of multimedia retrieval in which the core contributions of this thesis are settled. Before each of them is discussed in depth, a big picture of multimedia retrieval in general will be given. For this purpose, a basic terminology will be specified and the main concepts of the superior research field along with basic retrieval paradigms highlighted. Finally, it will shed light on most important research issues in multimedia information retrieval.

2.1 Terminology

The aim of this section is the definition of a basic terminology used within this thesis along with clear semantics and a focus on multimedia. The term media along with its categorization is defined ambiguously in literature. In this thesis media is used to define the physical media, such as a DVD, or person delivering information. This terminology follows the definitions of Schmitt [Sch06].

Term 1 (*Media*)

A specific form of digital data used to store and deliver information.

Besides this, media can be categorized by lots of ways [MW03, p. 33-34], e.g., by a human method of perception (visual or audio) or timing dependencies (discrete vs. continuous media).

Term 2 (*Media type*)

A specific taxonomy of different characteristics of a media in the domain of computer science.

A widely used taxonomy is the following: audio, (vector) graphics, image, text or video. A consideration of different media types as well as their facets are not in the scope of this thesis, but can be found in [Ste99] and [Hen03]. Using media and media type, multimedia can be specified:

Term 3 (*Multimedia*)

A set of media files stemming from different media types.

A specific concept or a real life object, such as the portrait of the Mona Lisa, can have several physically different multimedia resources carrying the same semantic, e.g., a digital image and a thumbnail of the Mona Lisa. This leads to the fact that a particular object may be related to different (multimedia) resources:

Term 4 (*Resource*)

An (additional) instantiation of a specific media, e.g., thumbnail of an image.

When talking of resources, the term metadata is frequently used in literature [Nat04]:

Term 5 (*Metadata*)

Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information.

In the domain of information retrieval and the Web, the term document is frequently used [Sch06]:

Term 6 (*Document*)

A logically connected and digital portion of text.

Term 7 (*Multimedia document*)

A document where its atomic data parts consist of different media types.

The yet introduced terminology serves as a foundation for the rest of the thesis. It will be extended by specific terms in the particular chapters.

2.2 Excursus: Information Retrieval

Since multimedia information retrieval is related to *information retrieval* [Sin01, APC05], this section gives a brief overview of the underlying research field. Before introducing its characteristics, a basic definition of Manning et al. [MRS08] shall be given:

Term 8 (*Information retrieval*)

Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

This definition clearly shows the difference to relational database management systems [Cod70]. In traditional databases, documents are retrieved that satisfy the condition of a precise query. Thereby, a query evaluation results in an unordered set of documents. In literature this is often called *data* or *exact retrieval*.

In contrast to that, information retrieval is intended to handle uncertain semantics of documents as well as vague specifications of the users information needs. Manning et al. [MRS08] defines *relevance* as a central term:

Term 9 (*Relevance*)

A document is relevant, if it is one that the user perceives as containing information of value with respect to their personal information need.

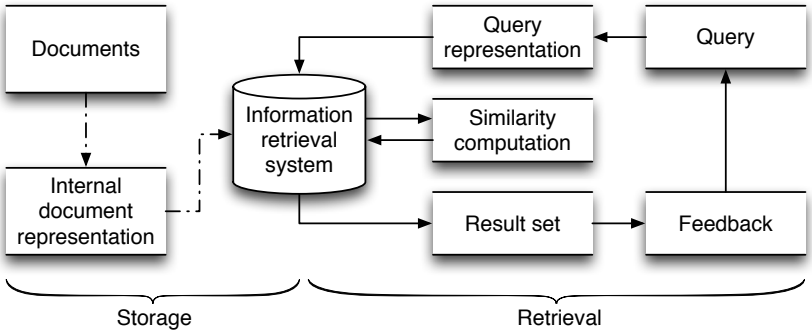


Figure 2.1: Abstract information retrieval system

With this knowledge an abstract information retrieval system[6] with its two main operation cycles, namely *storage* and *retrieval* (see Figure 2.1) are considered in literature. The most basic one is to save data in the retrieval system. Beside the actual storage of each document, an internal document representation is generated that describes the semantic information encapsulated in the document. This internal document representation is the basis for the retrieval process. The starting point of the retrieval process is the formulation of an initial query following a users information need in an non-precise way. The query is translated into an internal representation leading to a similarity computation, which is performed by comparing the generated document representation with the internal stored document representations. The outcome of this process is a set of documents, sorted with its relevance to the query in descending order. This procedure is called *similarity search* or *query by example*. Each document is part of the result set that achieves a certain threshold. Without this limitation, all stored documents would be part of the result set, since irrelevant documents are labelled with a very low relevance value. Due to the present uncertainty, also irrelevant documents may be marked relevant. To overcome this issue, the user is able to iteratively improve or refine the query. This can be done manually by the user or semi-automatically by the system, called *relevance feedback* [BYRN99]. Further, the term *ad-hoc retrieval* is frequently used in literature for systems that do not support feedback functionalities between different query sessions and operate on a fixed set of documents. In contrast to that, the term *routing* is used in systems, that forward queries to specific expert systems[7] for a certain domain [Har95].

[6] In this chapter only a high level overview of information retrieval is given. A detailed consideration of the workflow, e.g., metadata modelling, feature extraction or similarity computation with a special focus on multimedia, is part of the following chapters.

[7] An expert system is an artificial intelligence based system that is able to answer questions bound to the knowledge of a specific domain with high accuracy.

In this sense, information retrieval workflows using an automatic extraction of information from (multimedia) documents to leverage the overall retrieval process are referred as *content-based (multimedia) information retrieval* [Mit06].

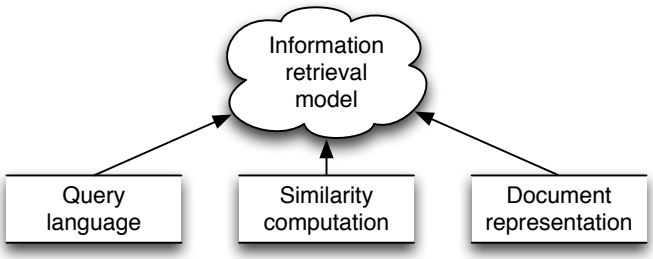


Figure 2.2: Components defining an information retrieval model

As the abstract information retrieval system implies, the most essential components specifying a retrieval model are the query language, the similarity computation and the (internal) document representation, as shown in Figure 2.2. Several different models have been proposed in the literature. In the following, three models with a direct correlation to this thesis will be introduced: *Boolean model*, *fuzzy model*, and *vector-space model*. Since most of the models have been evolved for text retrieval, the following basic terms follow textual naming convention, but can be applied to any other media types without loss of generality [MRS08]:

Term 10 (*Term*)

A term is a (perhaps normalized) type that is included in the information retrieval system's index.

Term 11 (*Index*)

An index is a structure containing all terms enabling efficient access in order to avoid linearly scanning of documents at retrieval time.

In this context, one or more terms serve as the data representation and a numeric weight is assigned to a specific term document pair. This weight is a measurement for the distinction of the document with respect to the overall set of documents by the given term. Term weight should not be set in conjunction with *term frequency* and *inverse document frequency (tf*idf)* [SB88] in the first place since those would need further semantics and definitions.

In the following, three representatives of information retrieval models will be introduced with a focus on their specific abilities:

Boolean model builds on top of set theory as well as Boolean logic [Bel05]. Following this, a query consists of a set of terms, which can be connected by Boolean operators. Those are the binary operators `AND`, `OR` and the unary operator `NOT`. The similarity computation in this case is equal to a lookup, whether the term is absent or present in the document, leading to a binary

term weight. This is modelled by `0` for an absent and `1` for a present term. In literature this is often called a *crisp set*[8]. The evaluation of the Boolean operators is as follows: `AND` is interpreted as intersection, `OR` as union and `NOT` corresponds to complement.

Beside the classical Boolean operators, there exist several extensions, such as `BUT` [Sch06] (combination of `NOT` and `AND`) or `PROXIMITY` [ANS95]. The latter defines the spatial closeness between two terms in the text to be identified as relevant. In this sense, closeness is a configurable, numerical value describing a word distance interval.

The Boolean model is a very elementary model with a logical basement and clear semantics. However, it exhibits two major drawbacks. First, the binary term weight reduces the best match procedure of information retrieval to exact retrieval of common database systems. The second drawback is the logical consequence of the first. Due to the fact, that all relevant documents are marked with `1`, a ranking with respect to the relevance of an individual document is possible, but useless.

Fuzzy model can be seen as an extension of the Boolean model to enable similarity search. An introduction of the basic fuzzy logic concepts along with the used terminology can be found in Appendix A of this thesis. By the help of gradual transitions offered by fuzzy sets[9], it is possible to express uncertainty and this therefore improves the aforementioned limitations of binary evaluation exhibited by crisp sets. In the fuzzy model, all stored documents serve as the universe and a term characterizes a document by a membership grade defined by the fuzzy set. A query is formulated in the same way as in the Boolean model by the use of Boolean operators. For the retrieval tasks, specific generalizations of the classical Boolean operators are in use, namely fuzzy complement, fuzzy intersection (t-norm) and fuzzy union (t-conorm)[10]. The use of the *standard fuzzy set operations*[11] often promotes *single value dependencies* due to the use of `MIN/MAX` that take only the minimum or the maximum membership grade into account [LKKL93]. Besides this issue, the result set of similarity search consists in principle of all documents available in the universe. Several techniques, such as a threshold describing a minimal membership grade, can be applied to shrink the result set to a suitable size.

Vector-space model proposed by Salton et al. [SWY75] utilizes relevant concepts of linear algebra [Str09] for internal document representation and query formulation. Here, a vector represents a document. Each dimension of the vector represents a term defined in the fixed, sorted index and every document holds a specific term weight for every term of the index. In contrast to the Boolean model, the vector-space model is able to handle similarity search

[8] Cp. Appendix A, Definition 15
[9] Cp. Appendix A, Definition 16
[10] Cp. Appendix A, Definition 20 to 22
[11] Cp. Appendix A, Definition 17 to 19

without further extensions. A query consists of a document represented by a vector and the similarity computation is calculated between two vectors. The two vectors can be seen as specific points in the spanned vector space, in which a similarity or distance function computes the actual degree of matching. A comprehensive overview of similarity and distance functions is given in Chapter 4.3.

As already mentioned, there exist even more models, such as the probabilistic model [CS10]. This model is based on probability theory defining the likelihood of a certain pattern recognized in a document, the membership of a document to a certain cluster as well as the relevance of the document to a user's needs. A further consideration of probability theory and therefore of the probabilistic model is not part of this thesis.

2.3 Classification of Multimedia Information Retrieval Techniques

Beside the variety of retrieval models, the research community has developed many approaches and techniques to improve multimedia information retrieval in a large variety of application domains [EABC+11, WBDB+06, ST07]. For overview reasons, a few categorizations have been issued. This thesis will follow the classification issued by Rüger on the basis of the utilized query types [Rue10]:

Piggy-back text retrieval performs a full text search over a set of unstructured strings. These have been extracted automatically in a pre-processing step from the multimedia data stemming from closed-captions of videos or derived by a speech to text analysis of audio data. Further details on this query category are not in the scope of the thesis.

Metadata-based retrieval operates on a set of documents that are structured by the use of a specific metadata format. In this sense, a metadata format stores data in a semantically enriched as well as computer understandable way. Due to the well-structured nature of metadata formats this fosters the accuracy of retrieval engines. A closer inspection of metadata schema is part of Chapter 3.

Content-based retrieval uses information encapsulated in the content of the media resource. There exist several techniques to describe the content of a media resource. Those descriptors of a media resource are called *features*, which are covered by Chapter 4. In the multimedia domain, the characteristics expressed by a feature vary for example from colour over shape to texture. The input data for such a query is mostly a media resource, such as an image for which similar images shall be retrieved. This technique is called *query by example*, but there exist a few more, such as *query by feature* or *query by sketch* [SKS10].

These query categories are intended to be the most possible abstraction layer to be suitable for all techniques and approaches. Obviously, there exist proposals that combine different query types to reach higher retrieval accuracy, such as the combination of content-based and metadata-driven retrieval [ACB02].

2.4 Multimedia Information Retrieval: An On-going Challenge

Taking a short look into history, the research community started elaborating on efficient and user-friendly multimedia information systems already in the 1980s [HLMS08]. From there on, the evolution of multimedia information retrieval passed through two major stages. The first stage lasted until the mid 1990s resulting in the definition of basic feature families for describing media resources as well as methods to enable machine-level indexing and retrieval of multimedia documents. The second stage continues today and deals with high-level semantics. Here, the covered scenery or setting of a media resource shall be extracted to be ideally coherent with a humanÂ´s perception. Within this process, several specific sub-research fields have been formed focusing on specific multimedia types, such as images [DKM09].

The human perception as well as the way content of media resources can be interpreted leads to imprecision and subjectivity. Every human, depending on his social or cultural background, has his or her own perception with respect to the actual meaning of a media resources content. Along with this, only the context defines the semantically meaningful parts of a media resource.

In this light, Smeulders et al. [SWS+00] defined (among others) two *gaps* that are present in all facets of multimedia information retrieval. The first to mention is the sensory gap:

Term 12 (*Sensory gap*)

> *The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene.*

With respect to the topics covered by this thesis, the sensory gap is not in central focus (cp. Section 1.3). In contrast to that, the semantic gap is omnipresent:

Term 13 (*Semantic gap*)

> *The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.*

Figure 2.3 illustrates both gaps and their correlation in a very simplistic way.

A real world scenery, here a flag in front of a mountain in the Monument Valley, is somehow captured into a media resource, e.g., by an image sensor. Besides the different capture devices, e.g., camera vs. camcorder, the conditions while taking

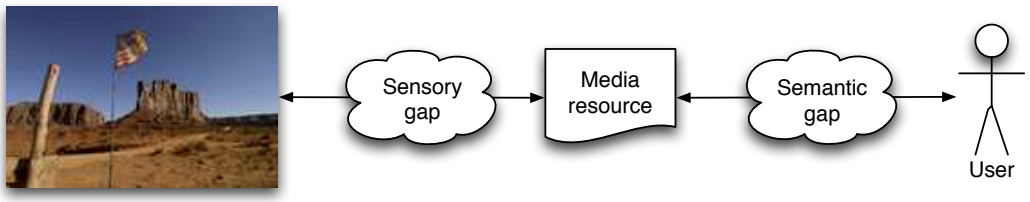


Figure 2.3: Correlation of the sensory and the semantic gap

the picture may vary, e.g., illumination or viewing angle. Summing up these influences, the sensory gap mostly yields to uncertainty which hinders disambiguation for example during a retrieval process. After capturing, the real life object is represented by a media resource. As already stated, in a multimedia information retrieval system certain features will be used during retrieval tasks instead of the actual media resource. In literature, one finds the term *low-level features* meaning an internal data representation that can be automatically computed from the multimedia content itself, e.g., a colour histogram. In contrast to that, the term *high-level features* express the scenery of a media resource, which is mostly generated and validated manually by humans. Obviously, the semantic gap is omnipresent between these two feature classes. Within the last decade, the term *mid-level features* [CG00] raised denoting the detection of objects in a multimedia resource, such as the detection of a persons face.

In the domain of still images, the semantic gap describes the issue that algorithms try to infer the actual meaning of a still image by the use of pixel data. Current research is still trying to improve and solve these gaps for very special domains, e.g., by expert systems, but a general solution for the semantic gap seems to be impossible. In this example, a regular user only sees the mountains of the monument valley in the picture, whereas a journalist may interpret the tattered flag as a symbol of the Indian drawbacks in society.

To get a broader view onto the semantic gap, it will be considered in terms of the multimedia life cycle [KBD+05]. Throughout this life cycle, metadata plays a central role between production, postproduction, and consumption, as illustrated in Figure 2.4 [SS06].

A multimedia resource regularly passes three main stages during its lifetime: *Creation*, *management*, and *transaction*. Each stage can be divided in canonical processes[12] [HON+08] forming an abstract processing chain, such as *search*, *annotate*, and *extract* for the *creation* stage. During these stages several different types of users are invoked, e.g., creator, producer, or consumer. Those users interact with the media resource and the corresponding metadata. The life cycle of a media resource tends to its transaction stage, where metadata information is exploited by the user for delivery, distribution, or sales. If the semantics of the media resource

[12]Hardman et al. define a canonical process as the most general description of a fundamental process to foster interoperability among different peers. A detailed consideration is part of Chapter 6.

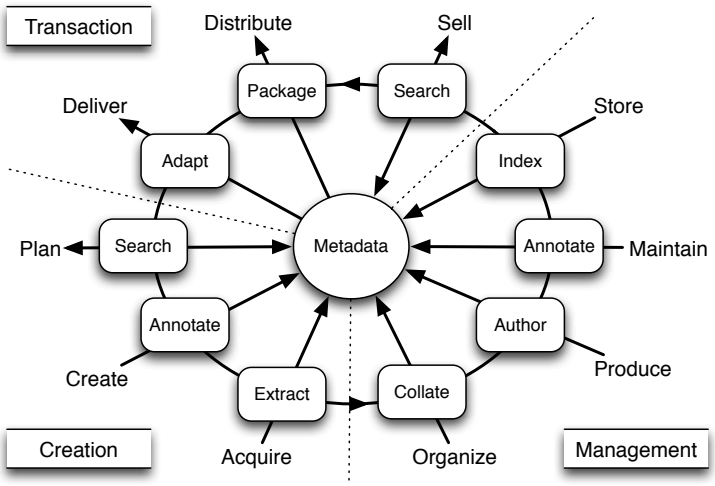


Figure 2.4: Multimedia life cycle and metadata

are captured in a misinterpreted or even falsified way due to the semantic gap, an efficient interaction with media resources cannot be established.

An overview of current research directions and challenges in the field of multimedia retrieval can be found in [Jai08] as well as [RHW10].

Chapter 3

Modeling Multimedia Metadata

The previous chapter introduced essential components of multimedia information retrieval. However, sophisticated workflows can only be guaranteed by the use of precise metadata [Nac00]. This observation is emphased by the presented multimedia life cycle, where metadata saturates all canonical processes, such as adaptation or search. Moreover, metadata formats enable an effective indexing of huge multimedia collections by harmonizing available information by a applying specific structure. Despite all advantages, diverse metadata formats also foster interoperability issues that have to be solved inside an application context.

This chapter aims to give a detailed summary on multimedia metadata. In the beginning, basic terminology is defined and the characteristics of multimedia metadata are recapitulated. Further, a categorization of metadata formats depicts their wide applicability and usage. After a consideration of meta models used for the creation of metadata schemas the different storage approaches are highlighted. Finally, standardization efforts along with nascent issues and ongoing trends in multimedia modeling conclude this chapter.

3.1 Only Data about Data?

Data modeling in general defines the process to setup requirements and to perform an analyzation on the integration of (real world) objects and their relationships into an information retrieval system. Following this, Santini [San06] defines *multimedia data modeling* as follows:

Term 14 (*Multimedia data modeling*)

> *Multimedia data modeling refers to creating the relationship between data in a multimedia application.*

Due to the complex and nested structure of multimedia data, object-relational design techniques are favored compared to semi-formal methods, which put relations into the center instead of the object itself [MPR+99]. Object-oriented approaches offer the flexibility to generate high-level abstractions as well as to capture the behavior of multimedia resources, e.g., timing or synchronization. The *multimedia metadata* definition of Bailer et al. [BBD+08] underlies its manifold characteristics:

Term 15 (*Multimedia metadata*)

> *Multimedia metadata describes various aspects of multimedia content, including formal and technical properties (e.g., encoding, format), information about the*

creation of content, the processing applied, its use, rights information and the structure and semantics of the content itself.

The definition of multimedia metadata implies various characteristics. A distinction for a single media resource is between global valid metadata information, e.g., creation of content, and metadata that changes over time, e.g., by the processing applied. In contrast to that *associative metadata* [Dun03] is connected to (potentially) more than one media resource. To ensure consistency and integrity, one has to be aware of update techniques in case of changes at the media resource or the multimedia metadata to ensure a consistent state.

Looking a little bit more into the content of multimedia metadata, *(multimedia) metadata semantics* come into play [ONH04]. In general, *semantic* describes the meaning of data, whilst *syntax* organizes its structure. It is essential to keep in mind, that the semantic meaning of metadata heavily depends on its context. In this sense, context could be defined by the application domain in which the annotated multimedia resource is embedded, interacting user groups or an associated cultural background. However, this few context examples clearly indicate that they are insufficient for metadata modeling since their blurred semantics leave room for misinterpretation. Metadata schemas, such as Dublin Core[13] or MPEG-7[14,15], exactly aim in reducing uncertainty by adding clear semantics to description elements, also called *properties*, along with a well-defined syntax for validation tasks [BBD+08]:

Term 16 (*Metadata schema*)

A metadata schema describes the semantics and value restrictions of description elements as well as relations between description elements.

Within metadata schemas, classification schemes or controlled vocabularies are frequently used to define explicit range of values [ANS05]:

Term 17 (*Classification scheme*)

A method of organization according to a set of pre-established principles, usually characterized by a notation system and a hierarchical structure of relationships among the entities.

[13] `http://dublincore.org/documents/dces/`, last visited December 18, 2013.

[14] `http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm`, last visited December 18, 2013.

[15] A further examination of standardized multimedia metadata schemas is part of Section 3.5.

Term 18 (*Controlled vocabulary*)

A list of terms that have been enumerated explicitly. This list is controlled by and is available from a controlled vocabulary registration authority. All terms in a controlled vocabulary must have an unambiguous, non-redundant definition. At a minimum, the following two rules must be enforced:

1. *If the same term is commonly used to mean different concepts, then its name is explicitly qualified to resolve this ambiguity.*
2. *If multiple terms are used to mean the same thing, one of the terms is identified as the preferred term in the controlled vocabulary and the other terms are listed as synonyms or aliases.*

As shown, metadata schemas offer great possibilities to establish a well-formed overall structure of metadata information. Further, classification schemes as well as controlled vocabularies are in use to explicitly add semantically enriched and uniform descriptions to specific entities of annotations. Typically, classification schemes and controlled vocabularies are created and hosted by major stakeholders of a given domain. In the broadcasting domain, the European Broadcasting Union (EBU)[16] is the major vendor for those documents, including definitions of user roles[17] or genres[18].

Besides all benefits of metadata schemas in enabling semantic descriptions of multimedia resources, the degree of semantic expressiveness differs from schema to schema. There exist metadata schemas focusing on unstructured information, such as a flat list of properties or key-value pairs as most minimal form. In contrast to that, other metadata schemas are highly structured with very complex hierarchies, where single properties have been split up in various atomic parts. This observation obviously leads to metadata interoperability issues present in various forms. In order to discuss the metadata interoperability issue in more detail, a definition from Haslhofer and Klas [HK10] is given:

Term 19 (*Metadata interoperability*)

Metadata interoperability is a qualitative property of metadata information objects that enables systems and applications to work with or use these objects across system boundaries.

Following this definition, interoperability issues of metadata schemas considered in this thesis concern the information level. Those issues can be further divided in *model-level* and *instance-level heterogeneities*. Model-level heterogeneities can be split in the following two categories:

Structural heterogeneity appears due to diverse structures in the corresponding data models. These diversities can be present on the one hand in element

[16] http://www.ebu.ch/, last checked December 18, 2013.
[17] http://www.ebu.ch/metadata/cs/ebu_RoleCodeCS.xml, last checked December 18, 2013.
[18] http://www.ebu.ch/metadata/cs/ebu_ContentGenreCS.xml, last checked December 18, 2013.

definitions itself (e.g., structure or naming) or in the applied domain representation. The latter is on hand, if the overall expressiveness of two models for the same concept are not equivalent, e.g., missing properties.

Semantic heterogeneity can be identified in terms of domain conflicts. Those contradictions can be found between schema definition languages (e.g., OWL vs. XML) as well as between correlation of metadata schemas stemming from different application domains. Further, homonyms or synonyms cause misinterpretations due to semantic ambiguities.

In contrast to model-level heterogeneities, instance-level heterogeneities are not subdivided in the two categories, they only belong to semantic heterogeneities. Further information on the topic of metadata interoperability can be found in [HK10].

(a) MPEG-7 (b) Dublin Core

Figure 3.1: Graphical excerpts of multimedia metadata descriptions

Figure 3.1 illustrates structural heterogeneity exemplary for the *creator* property, which is (amongst others) defined in MPEG-7 and Dublin Core. Let's consider the playwright "John Smith" should be described by metadata documents of both standards. The only possibility to model this piece of information in Dublin Core is to store it as unstructured string in the *creator* property. This is valid, since a playwright is a more specific concept than creator. In MPEG-7 exists also a *creator* property, but the name is split up in *family* and *given name* and a associated role gives further information. Here, the playwright role description could be taken from the EBU[19] classification scheme[20]. This example clearly shows, that MPEG-7 offers richer semantics than Dublin Core due to the loss of the specific information "playwright".

A special facet of multimedia metadata is the partition into *content-dependent* and *content-descriptive metadata* [Bim99]. In literature, one finds *low-level features* as a synonym for the first and *high-level features* for the latter. A detailed consideration of low-level is part of Chapter 4.

[19] http://www3.ebu.ch/, last checked December 18, 2013.
[20] http://www.ebu.ch/metadata/cs/ebu_RoleCodeCS.xml, last checked December 18, 2013.

Table 3.1: Usage scenarios of metadata schemas

Type	Description	Examples
Administrative	Administration and management of data collections, such as location of multimedia resources	METS[21], EAD[22], MODS[23]
Descriptive	Identification and retrieval of multimedia resources, e.g., content-based features or tags	Dublin Core, MPEG-7, XMP[24]
Preservation	Migration and annotation of data to ensure a long durability including provenance information	PREMIS[25], LMER[26], PRONOM[27]
Technical	Technical descriptions, e.g., formats, compression or security aspects	EXIF[28], ID3[29], QuickTime[30]
Usage	Specification of access and type of use, e.g., user management and versioning of media resources	MPEG-21[31], OGG[32], WebM[33]

3.2 Classifying Metadata Schemas

Similar to the manifold characteristics of multimedia metadata, different categorizations for proprietary as well as standardized metadata schemas can be applied [Dun03].

Table 3.1 illustrates a categorization, which is based on the following abstract usage scenarios of metadata schemas: *Administrative*, *descriptive*, *preservation*,

[21] http://www.loc.gov/standards/mets/, last visited December 18, 2013.
[22] http://www.loc.gov/ead/eadschema.html, last visited December 18, 2013.
[23] http://www.loc.gov/standards/mods/, last visited December 18, 2013.
[24] http://www.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMPSpecificationPart1.pdf, last visited December 18, 2013.
[25] http://www.loc.gov/standards/premis/v2/premis-2-0.pdf, last visited December 18, 2013.
[26] http://nbn-resolving.de/urn:nbn:de:1111-2005051906, last visited December 18, 2013.
[27] http://www.nationalarchives.gov.uk/aboutapps/pronom/pdf/pronom_unique_identifier_scheme.pdf, last visited December 18, 2013.
[28] http://www.exif.org/Exif2-2.PDF, last visited December 18, 2013.
[29] http://www.id3.org/Developer_Information, last visited December 18, 2013.
[30] http://developer.apple.com/mac/library/documentation/QuickTime/QTFF/QTFFPreface/qtffPreface.html, last visited December 18, 2013.
[31] http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm, last visited December 18, 2013.
[32] http://www.xiph.org/ogg/, last visited December 18, 2013.
[33] http://www.webmproject.org/code/specs/container/, last visited December 18, 2013.

technical and *usage*. Apparently those describe a very broad domain of possible applications, where each of them exhibit specific needs. Since those application domains share an (at least slight) overlap, a tight separation of the metadata schemas listed in the examples column is not possible. Therefore, some metadata schemas could be associated to more than one usage scenario due to their richer expressiveness, e.g., MPEG-7.

As already mentioned, a certain application domain has its own requirements that have to be reflected regarding multimedia metadata modeling. Due to this fact, a huge number of metadata schemas raised within the last decades resulting in a *metadata standards alphabet soup* [SS06]. To fit the requirements of complex application domains, domain dependent metadata schemas have been created. An example for this would be DICOM[34]. It has been designed for the medical imaging domain capturing essential steps of examination workflows, such as management, storage and transmission of medical analysis. Besides image related metadata, demographic data of patients is also accumulated within this standard. However, domain dependent metadata schemas on the one hand offer rich semantics but on the other hand favor interoperability issues in terms of information exchange between different peers. This is caused by misinterpretations on the instance level due to ambiguous or blurred semantics. In contrast to that, domain independent metadata schemas, e.g., Dublin Core, disclaim the use of expert knowledge in the definition of the underlying semantics, e.g., medical naming conventions, to foster a wider applicability and an decrease of interoperability issues.

Beside the membership to application domains, a metadata schema may be media agnostic. This means that the metadata schema is not bound to a specific media type, e.g., Dublin Core. Other metadata schemas are only suitable for specific multimedia resources. An example for this would be ID3, which is only used for the description of audio files.

3.3 Metamodels for Designing Metadata Schemas

Until now, the chapter focused on multimedia metadata and its correlation to metadata schemas. In order to discuss the possibilities how metadata schemas can be created, this correlation has to be expanded to a big picture by considering further abstraction layers. A suitable arrangement has been issued by the Object Management Group (OMG)[35] in the Meta Object Facility (MOF) specification [OMG11]. It defines the following four building blocks along with the relation to each other, as illustrated in Figure 3.2 [Poe06]:

M0 - Metadata is the lowest level. It is a specific metadata document and an instantiation of a metadata schema.

[34] `http://medical.nema.org/standard.html`, last visited December 18, 2013.
[35] `http://www.omg.org/`, last visited December 18, 2013.

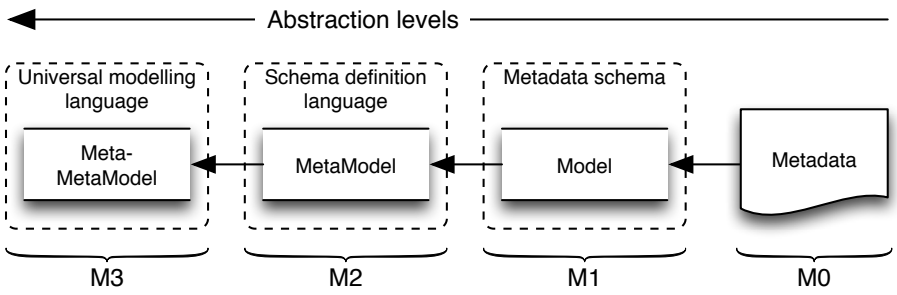


Figure 3.2: Abstraction levels of metadata building blocks

M1 - Metadata schema follows Term 16 of this thesis. It assigns clear semantics and structural information to a metadata document.

M2 - Schema definition language offers an abstract syntax to create a specific metadata schema. Essential components of this syntax are language primitive attributes and a formalism how they can be arranged.

M3 - Universal modeling language serves as the most general abstraction as well as a basis for schema definition languages. Two options can be used to define this layer: usage of a different modeling mechanism, e.g., an explicit set of axioms, or an own modeling approach, e.g., the MOF model.

The remaining will put the M2 layer into scope by analyzing metamodels commonly used to create metadata schemas. Here, we distinguish between representational models and multimedia ontologies. The final report [Hau07] of the W3C Multimedia Semantics Incubator Group[36] confirms this partition, since metadata schemas currently in use mostly rely on XML Schema or Semantic Web compatible languages. In the following those will be in the centre of discussions. Nevertheless, other metamodels are mentioned.

3.3.1 Representational Models

Representational models are not restricted to modeling purposes, but can be also used within applications to directly operate on object structures. The most prominent example for a flexible metalanguage is the Extensible Markup Language (XML)[37] [BPSM+08]. XML at its core is an enhancement of the Standard Generalized Markup Language (SGML)[38] [ISO86] with the aim to become the de-facto standard language for a platform independent data exchange on the Web. Basically, a XML document forms a hierarchical tree. This tree exposes specific characteristics, namely *labeled*, *boundless* and *ordered* [AMR+12]. This means a label (e.g., an

[36] http://www.w3.org/2005/Incubator/mmsem/, last visited on December 18, 2013.
[37] http://www.w3.org/XML/, last visited December 18, 2013.
[38] http://www.w3.org/MarkUp/SGML/, last visited December 18, 2013.

annotation) is directly added to a node and the according children are initially not restricted to their number, but follow a certain ordering.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <rootElement attr1="value1" attr2="value2" ... >
3   <nestedElement1>
4     Unicode content
5   </nestedElement1>
6   <!-- This element is empty! -->
7   <nestedElement2 />
8 </rootElement>
```

Listing 3.1: Basic example of a XML document

The central syntax of a XML document is illustrated in Listing 3.1. A document may start with a optional prologue specifying the version and the document encoding, in this example UTF-8 [Yer03]. Basically, a XML document consists of elements, which declaration is made of an *opening* and an *end tag*, c.p. line 2 and 8. In between, nested elements can arise, c.p. line 3 and 7, or simply a unicode text, c.p., line 4. An element can be empty and may be noted in an abbreviated form, as shown in line 7. Additionally, the opening tag can include attributes, c.p. line 2. In contrast to elements, attributes are not ordered, but their naming has to be unique within a single element. There is also the possibility to add comments to a XML document, see line 6. A XML document is *well-formed* if there exist a *root element* and the element tags close in the opposite order they have been opened. To avoid naming conflicts between ambiguous concepts, Uniform Resource Identifiers (URIs) [BLFM98] and XML namespaces [BHL+09] are used to generate a so called *markup vocabulary*. XML namespaces also foster modularization and re-use of XML names.

XML documents can be differentiated in two serialization categories. On the one hand, the *serialized form* is a linear representation of the text. It is used for example in communication protocols of the Web, such as the well-known web service implementation Apache Axis2[39]. On the other hand, the *tree-based form* offers an abstract representation of the tree. In terms of the Web, the Document Object Model (DOM)[40] has been standardized by the World Wide Web Consortium (W3C)[41] enabling an object-oriented document manipulation in its frameworks.

Since XML only defines the syntax, further techniques are needed to describe semantics. A first step into this direction was the Document Type Definition (DTD)[42] providing means for specifying constraints on the basis of regular expressions. A much richer approach to add (to some extend) semantics to XML documents is XML Schema[43] [FW04]. It uses XML syntax and defines the structure of a XML

[39] http://axis.apache.org/, last visited December 18, 2013.
[40] http://www.w3.org/DOM/, last visited December 18, 2013.
[41] http://www.w3.org, last visited December 18, 2013.
[42] The DTD definition is part of the already references W3C recommendation of XML.
[43] http://www.w3.org/XML/Schema, last checked December 18, 2013.

document in a top-down manner, such as naming, number or ordering of elements or type restrictions of attributes. However, XML Schema is not able to add machine-readable semantics to the meaning of the elements and therefore inference of new knowledge is a cumbersome task mostly performed by manual inputs or domain knowledge.

When speaking of representational models on the Web, one might consider the JavaScript Object Notation (JSON)[44] [Cro06] as another possibility to design metadata schemas. JSON is a platform independent and lightweight alternative to XML with a very low (technical) overhead. Due to its overall aim to be a programming language and data interchange format it is not designed as a document model nor as a markup language. Therefore it is not suitable for the creation of metadata schemas.

3.3.2 Multimedia Ontologies

Within the last ten years, the Web reinvented itself over and over, which led from a more or less static and silo based Web to a open Web of data, the so called Semantic Web[45,46] [BLHL01]. The main intention of the Semantic Web is to provide an open accessible, machine-readable and semantic description of content by the use of ontologies. Those are frequently used to model multimedia metadata information. The Resource Description Framework (RDF) [KC04] is the basement of the Semantic Web by providing a formal language to represent structured information without loosing the semantic information of the underlying data. It can be seen as an enhancement of XML. In contrast to the tree based structure of XML documents, RDF documents form at least one directed graph structure [HKRS08]. Those graph structures consist of nodes and edges, which are labeled with URIs. The edges of the graph describe the semantic relationship between the nodes. The frequently used term *triple* is based on the way, this graph is generally represented. In computer science there exist several ways to describe a graph structure, for example by a adjacency matrix. Due to the fact that RDF graphs are usually sparse, the cells of a matrix would be rather empty leading to an inefficient representation. A triple $< s\ p\ o >$ is composed of a *subject* s, *predicate* or *property* p and an *object* o meaning an object o is a value of property p for the subject s. In RDF, data values are called *literal* and are only valid in place of an object. RDF data can be encoded in various formats, such as Notation 3 (N3)[47], N-Triples[48], Terse RDF Triple Language (Turtle)[49] and RDF/XML[50].

An example for a RDF serialization in Turtle is shown in Listing 3.2 describing that a thesis is written in English by the postdoctoral researcher John Doe. The

[44] http://json.org/, last visited December 18, 2013.
[45] http://www.w3.org/standards/semanticweb/, last checked December 18, 2013.
[46] This section is based on parts of [SGD+09].
[47] http://www.w3.org/DesignIssues/Notation3.html, last visited December 18, 2013.
[48] http://www.w3.org/2001/sw/RDFCore/ntriples/, last visited December 18, 2013.
[49] http://www.w3.org/TeamSubmission/turtle/, last visited December 18, 2013.
[50] http://www.w3.org/TR/REC-rdf-syntax/, last visited December 18, 2013.

```
1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
3 @prefix dc:  <http://purl.org/dc/elements/1.1/> .
4 @prefix ex:  <http://ww.example.org/> .
5
6 ex:SomeThesis    dc:langauge   "en-US"^^xsd:string ;
7                  dc:creator    ex:John_Doe .
8 ex:John_Doe      rdf:type      ex:Person , ex:PostDoc .
```

Listing 3.2: Basic example of a RDF document in Turtle syntax

first part of the RDF document is declaring the namespaces that are used in the document. The namespaces declare a vocabulary composing a set of identifiers with a specific semantic meaning. In this example, the namespace `rdf` defines the syntax of the RDF document and a class membership, c.p. line 8, whereas the namespace `xsd` offers possibilities to type literals with a data type, c.p. line 6. Despite structural information, the namespace `dc` integrates the Dublin Core metadata schema. Here, it adds a semantic meaning to the data by expressing the language of the thesis as well its creator. The remaining namespace `ex` serves as a example namespace holding further information for the resources thesis, John Doe and Person. The example also shows a abbreviation in line 7. Here, the subject from line 6 is passed on to the next line by the semicolon. A similar form is shown in line 8, where different objects for the same subject-predicate pairs are summed up.

The URIs used in RDF documents need not to be existent, but should be only used for abstract resources in that case. If an URI is existent, it offers in general more information about the represented concept in several instantiations, e.g., serializations or languages. A user client is then able to choose the designated resource. This procedure is called *content negotiation* [HM98] and stems originally from the HTTP protocol.

Until now, the capability of semantic expressiveness of RDF has improved in comparison with XML, but machine-readability and interpretability of relations are primary enabled by ontology languages [AH08], such as RDF Schema (RDFS) [BG04] or Web Ontology Language (OWL) [MH04].

RDFS is the foundation to describe terminological knowledge between objects defined in a RDF document. Here, the supplied semantic expressiveness is limited by its coverage of set theory. Beside already available concepts of the RDF specification, e.g., `rdf:type`, RDFS offers language constructs to define an in-depth vocabulary, e.g., by `rdfs:Resource` or `refs:Class`, specific hierarchies, e.g., by `rdfs:subPropertyOf` or `rdfs:subClassOf`, or valid ranges. However, one is not able to address restrictions on class memberships or cardinality constraints with RDFS. OWL addresses exactly this problem, since it is based on the axioms of first–order logic [And10] enabling sophisticated reasoning tasks. It is a well known fact, that first order logic at its core is undecidable [HWZ02] and so is OWL, if

it relies on the whole set theory. To avoid this issue, three sublanguages with the following subset relation have been defined: OWL lite $\subseteq$ OWL DL $\subseteq$ OWL Full. OWL Lite as well as OWL DL are derived from specific description logics that ensures decidability. Here, OWL Lite utilizes $\mathcal{SHIF}^{(D)}$ that enables complements, transitive roles, role hierarchies, inverse roles, functional roles and concrete data types. In contrast to that, OWL DL relies on $\mathcal{SHOIN}(D)$. This description logic enlarges the aforementioned by nominals and number restrictions. OWL full comes without any restrictions and therefore its ontologies are undecidable. Due to modeling as well as syntactical inconveniences, OWL has been revised by the W3C leading to OWL 2 [HKP+09]. OWL 2 is based on the $\mathcal{SROIQ}^{(D)}$ description logic, fully backwards compatible and offers sublanguages similar to the initial version of OWL, here called *profiles*. Further details as well as an comprehensive overview of applicable description logics for the OWL language family can be found in [Rud11].

Topic maps[51] [Pep02] are another way to model knowledge using semantic networks. The semantic network is spanned by the three dimensions, often called *TAO*: *Topic*, *occurrence* and *association*. Here, topics are a set of subjects that can be any (real world) objects. Occurrence donates the linkage between a topic and a set of information resources. In topic maps, semantic is modeled by describing relations between topics named association.

3.4 Presence of Multimedia Metadata

Along with its diverse characteristics and modeling aspects, multimedia metadata may arise in different forms during an application context. Figure 3.3 shows three most essential layers that should be taken into account, namely *storage*, *transmission* and *presentation layer*. This layers follow the well known three-tier architecture approach. In the following, the most common appearances in the specific layers will be discussed with a specific focus on the Web.

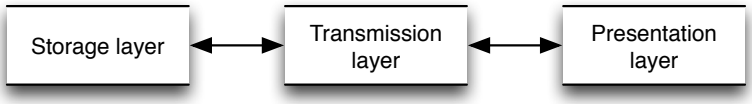


Figure 3.3: Accessibility of multimedia metadata information

Storage layer. A persistent storage of multimedia metadata is crucial for every application domain. Frequently, three storage approaches are in use in terms of multimedia metadata. The first utilizes a specific data management system for storage, such as a database or a information retrieval system. In such systems, the metadata information is split up in its atomic pieces and stored in an internal format and is physically detached from the media resource but logically linked. The

[51] http://topicmaps.org/, last visited December 18, 2013.

main advantages are secure management of transactions and fine granular, efficient retrieval due to index structures. In the second, the metadata information is stored together with the actual media resource in a *(multimedia) container format*. Here, a digital container format specifies a meta-file format organizing the coexistence of various media resources along with multimedia metadata, such as OGG or WebM. The physical combination of both eases updating issues and preparation for delivery. The last approach is a simple storage in the file system. Here, the metadata information is stored in dumps consisting of single documents, e.g., written in XML or RDF. The disadvantages are obvious in terms of retrieval abilities, update issues or delivery aspects.

Transmission layer. The delivery of multimedia metadata heavily depends on the application domain. Inside a closed application, with no present network transmission, multimedia metadata will be shipped by internal object structures, such as the already introduced DOM model. If network transmissions are present, the data delivery depends on the actual network protocol in use. Examples of the huge variety of possible protocols are binary transmission protocols, e.g., FTP[52], or other data exchange protocols, such as XML or JSON.

Presentation layer. During the evolvement of the Semantic Web, a specific usage of metadata in the presentation layer occurred. *Rich snippets* are frequently used to embed metadata and therefore semantics into user interfaces, mostly websites. Today, this term is frequently used for the way, Google result items are displayed to the user, as shown in Figure 3.4. In this example, a user searched for an Italian restaurant in New York. Beside presentation of the link or the title of a result item, additional information is shown to the user. In this example, accumulated user ratings as star schema as well as a small description of the restaurant are part of the result.

Figure 3.4: Example rich snippet of a Google result item

The prompt of a rich snippet is only a small part of those embedded semantics. In particular, it offers structuring and filtering purposes and is exploited by search engines to optimize the retrieval process to force a better page rank, summarized in the buzz word *search engine optimization (SEO)* [DD11]. For integration of metadata, three major approaches are applied:

[52] File Transfer Protocol

Microdata [53] has been driven by WHATWG[54] and is the standardized way to include semantic information in HTML5 documents. Microdata itself is organized in various items, which can be seen as a specific group. A group defines a set of various properties that made of key-value pairs. The most famous example of microdata is *schema.org*[55]. In this activity, major search engine vendors agreed on a specific markup, which is exploited during the search process.

Microformats [All07] constitute a set of open data formats that base on already well-known standards with a broad support of already available tools. Its central aim is to solve a specific issue, to be as simple as possible and to be able to reuse already defined formats. A differentiation is made between elemental, e.g., XOXO[56], or compound microformats, e.g., hCard[57]. In HTML the properties of a specific microformat are directly embedded by the use of the `class` tag in elements of a HTML document.

RDFa [AHSB12] was issued by the W3C and stands for Resource Description Framework in attributes. It extends HTML by a set of attributes, e.g., `about` or `property`, in order to integrate RDF data into XML-based documents. With this technique, fragments of already available ontologies, e.g., Dublin Core, can be used for a semantic document description.

Table 3.2 summarizes the pros and cons of the three formats. The decision, which format should be chosen for an application heavily depends on the usage scenario.

3.5 Metadata, Standardization and the Web

Recapitulatory, with the proliferation of media resources, the multimedia community accentuated the central role of metadata to describe media resources as well as to establish high-quality multimedia information retrieval systems [Nac00]. This finding has been reflected in the creation of a multimedia life cycle spanning over media resources, metadata and the user [KBD+05]. The main idea was to identify essential stages of multimedia resources between production and consumption. This led to a better understanding of multimedia and its interaction possibilities to align and refine workflows. Obviously, metadata standards on the one hand enable interoperability to interchange information about media resources between different stages and peers of the life cycle [SS06]. However, the major drawback is the aforementioned large variation of multimedia application domains hinders the adoption

[53] `http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html`, last visited December 18, 2013.
[54] `http://www.whatwg.org/`, last visited December 18, 2013.
[55] `http://www.schema.org`, last visited December 18, 2013.
[56] `http://microformats.org/wiki/xoxo`, last visited December 18, 2013.
[57] `http://microformats.org/wiki/hcard`, last visited December 18, 2013.

Table 3.2: Pros and cons of microdata, microformat and RDFa

Type	Pros	Cons
Microdata	+ native HTML5 + native JSON export + Microdata DOM API + search engine support	− no multiple property values − extends HTML
Microformat	+ no HTML extension needed (HTML4 compatible)	− usage of `class` may conflict with CSS definitions − no specified extraction API − no internationalization supported
RDFa	+ full flexibility + RDF Dom API specified + allows mashups of vocabularies	− high complexity − extends HTML

of a single, universal metadata standard. Hardman et. al. [HON+08] focused on this lack of harmonization at the interface level. They lifted the initial idea of the life cycle to canonical processes of semantically annotated media production. In their work, a canonical process is defined as the most general description of a fundamental process to foster interoperability among different peers. These were created with much attention and feedback of the multimedia community and serves as a foundation for further models. On the basis of the life cycle and an examination of multimedia systems, nine fundamental processes have been investigated: premeditate, create media asset, annotate, package, query, construct message, organize, publish and distribute. In all of them, metadata is omnipresent and essential for guaranteeing efficiency and quality. In this sense, focusing on the data modeling level a core vocabulary is not yet another metadata format, but a technique to ensure that the information exchange among different metadata formats used by (canonical) processes or systems follow standardized and clear semantics. Such standardization effort can be directly injected into canonical processes in order to improve interoperability with respect to data exchange.

Data exchange and thus metadata interoperability is an important task for both real-world application scenarios introduced in Section 1.1. In the medical as well as the image retrieval scenario, media resources and their annotations are an integral part, but unfortunately commonly applicable interaction techniques are still limited [ONH04]. The Semantic Web introduced concepts such as ontologies that were intended to improve the issue of unified media resource description. Within this, already existent and well-known metadata standards have been lifted into ontology description logics to enhance the semantic expressiveness and finally improve interoperability. Taking MPEG-7 as a representative, there were large efforts in the

translation of the XML schema version of the standard into an ontology. Troncy et al. [TCL+07] compared four different MPEG-7 ontologies in terms of coverage and scalability. In contrast to that, Tzouvaras et al. [TDMR+05] discussed MPEG-7 ontologies with respect to their impact on interoperability. Despite the MPEG-7 standard, such efforts have been widely applied to non-ontology metadata formats leading to an explosive spread of multimedia ontologies, as reflected in the survey of Suàrez-Figueroa et al. [SFAC11] and the already introduced report of the W3C Multimedia Semantics Incubator Group.

However, the transition of a metadata format into a Semantic Web compliant language does not have an additional benefit with respect to the overall semantics, in case of MPEG-7 the reduction of complexity. In contrast to that, ongoing activities focus on the creation of core vocabularies that clearly add semantic information. Such an approach will be introduced in Chapter 6 as a contribution of this thesis. It consist of a pivot metadata format and an API to enable alignment of metadata formats among each other.

Chapter 4

Indexing Multimedia Resources

So far, the chapters defined a basic multimedia retrieval scenario as well as insights on multimedia modeling leading to the impact of representational models and multimedia ontologies. Those have been defined as content-descriptive or high-level features. In the context of this thesie, it is essential to consider content-dependent or low-level multimedia features since they are utilized in the proposed multimedia fusion approach of Chapter 11. In general, those data structures are designed to create a compact representation over various media types and therefore enable (to some extend) similarity calculations. In contrast to manually created high-level semantics of a media resource, multimedia features stem from automatic extraction routines.

This chapter highlights different dimensions of multimedia resources to extract multimedia features and gives a categorization in diverse annotation levels constituting the indexing pyramid. An usage analysis of multimedia features in general is the basement for a taxonomy in the domain of visual multimedia features. Following this, an overview of adequate similarity measures will be given, which are utilized by similarity computations. Finally, the chapter introduces specific multimedia indexing structures, which improve the *curse of dimensionality* issue.

4.1 Usage of Multimedia Features During Retrieval

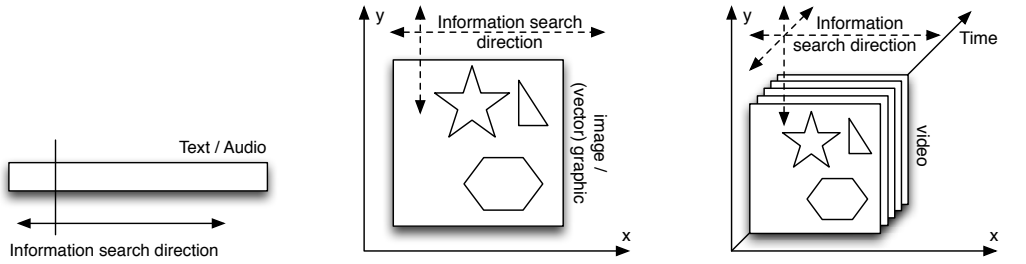
The process of indexing multimedia resources for retrieval heavily depends on the media type. The media type itself defines the actual dimensions, in which a content-dependent analysis of a multimedia resource can be conducted. With reference to the selected taxonomy in Section 2.1, Figure 4.1 assigns the following access dimensions to audio, (vector) graphics, image, text and video [Pra97]:

1-dimensional access is present in audio and textual data. Here, the data is mostly treated as a continuous stream along a single dimension, such as time or reading direction.

2-dimensional access is given in (vector) graphics and images. The two dimensions are spanned by the spatial relation inside those multimedia resources.

3-dimensional access exists in videos. Since videos are a mixture of the prior discussed media types, the spatial and temporal dimension get aggregated.

Besides these three conventional dimensions, multimedia container formats enlarge this set:



(a) 1-dimensional access (b) 2-dimensional access (c) 3-dimensional access

Figure 4.1: Dimensions of multimedia ressources

4-dimensional access occurs, if more than one multimedia resources are arranged in a surrounding one, e.g., in a multimedia container format. Here, the fourth dimension is the selection of the actual multimedia resource(s).

Obviously, there is a strong correlation between the defined access dimensions and a multimedia feature. In this regard, the actual representation of a multimedia resource together with a specific *multimedia segment* serves as the input parameter for a *multimedia feature function* producing the actual *multimedia feature vector*. The segmentation algorithms specifically applied in the image domain can be classified in the following categories [Jae05]:

- *Pixel-based segmentation* is the most basic way to segment an image by considering its illumination. Here, all color occurrences are reduced to grey values of the individual pixels. Depending on the actual quantification into n grey values, those algorithms lead to a n-ary histogram.

- *Region-based segmentation* can be seen as an enhancement of pixel-based segmentation. Instead treating a pixel as an isolated piece of information, it takes advantage of the following fact: objects enclosed inside an image exhibit a specific neighborhood connectivity between near pixels in terms of its color gradient. In principle, a clustering algorithm expands or merges the region around a pixel to find segments inside an image.

- *Edge-based segmentation* is more robust towards a illumination bias by searching for the border of an object. Most algorithms are divided in two phases: at first the image is scanned line by line for maxima in the gradient to detect edges. In the second phase, if a maximum has been detected, it is followed in order to detect the complete edge. Those two phases are repeated until the complete image is scanned.

- *Model-based segmentation* tries to detect several segments in an image by the usage of geometric shapes. In recent works, those shapes are not fixed, but will be adapted during processing phases.

An consideration of specific segmentation algorithms can be found in survey articles, such as [HS85] and [FM81]. In literature, a multimedia segment stemming from a partitioning of the input media resource is often termed *multimedia fragment* [TMPD12].

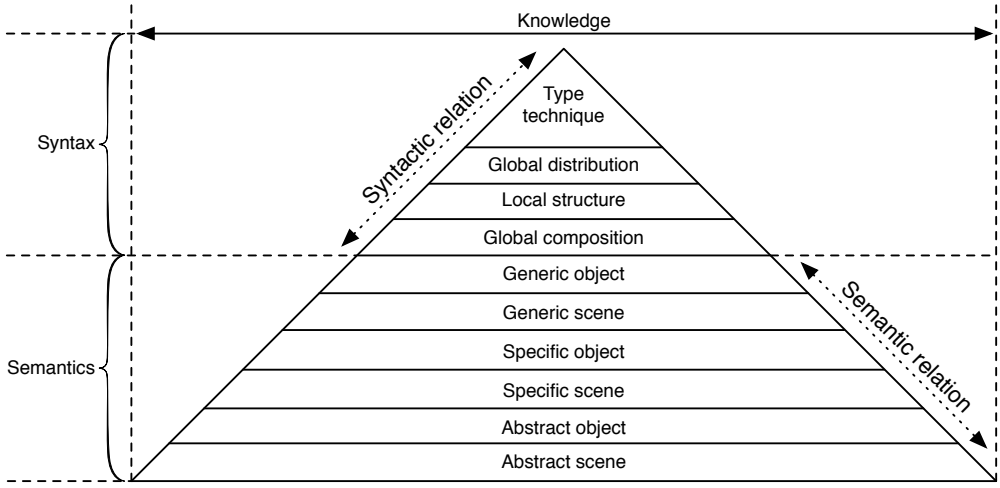


Figure 4.2: Indexing pyramid for classifying content attributes

Multimedia features in general expose very diverse characteristics and possible usage domains. In order to have an acceptable categorization of visual multimedia features, Jörgensen [JJBC01] proposed the indexing pyramid as illustrated in Figure 4.2. Basically, the pyramid is divided into two segments that specify whether a feature is capable for describing syntactical information or if it is possible to embed semantics as well. Apparently, this implicates an information gain from the top of the pyramid to its basement. For a finer granulation the two segments are partitioned in ten subparts. The lower subparts of the pyramid encompass high-level semantics such as meaning of a scene or an object. As already mentioned, the semantic levels can be hardly reached with single automatic routines and are processed by user inputs or feature fusion techniques. In contrast to that, low-level processing (first four pyramid levels) is the foundation of the pyramid and therefore the entry point for deeper analysis. Here, an essential differentiation is made between features characterizing *local* or *global structures* leading to a description of the *global composition*.

The large variation of available multimedia features makes it impossible for an application to take all of them into account. Especially in multimedia retrieval, a feature selection process is conducted to choose a specific set of multimedia features ensuring a sophisticated indexing on the basis of prior accomplished requirement analysis. Candan and Sapino [CS10] name a few reasons why a feature should be selected. First, *application semantics* have a strong impact. There might be multimedia features, which lead to higher accuracy in specific domains. In contrast to that, some features might favor the *perception impact*, e.g., motion sensitivity. Fi-

nally, multimedia features vary also in their discrimination power. This means some multimedia features foster similarity calculations during retrieval, e.g., in terms of face recognition. In this thesis, multimedia features are used in the *late multimedia fusion approach* presented in Chapter 11. Currently, only syntactic multimedia features are integrated, but the use of semantic features is envisioned.

4.2 Expressiveness of Features in Image Retrieval

As already mentioned, the landscape of multimedia features distinguishes especially in the way of describing specific facets of the multimedia content. Amongst others, Zhang et al. [ZIL12] propose the following classification to differentiate between the expressiveness of (syntactical) visual features in the domain of image retrieval:

Color features. Since color is an essential facet of an image and furthermore very important for a humans perception, it gained wide interest in the creation of multimedia features. Color itself can be specified by the use of *color spaces* [GW01], such as RGB, CMYK or HSV color space. RGB is an additive color space used in common electronic display devices, which means it mixes three basic color tones, red, green and blue, to produce the desired color. In contrast to that, CMYK is a subtractive color space used in the print sector, where cyan, magenta, yellow, and key (black) are used as basis. HSV stands for hue, saturation and value and is nearest to the humans perception. On top of the color spaces, a plethora of multimedia features have been developed covering color-based descriptions of an image. Two very simple color-based multimedia features are *color moments* [FSA+95] and *color histogram* [SB91]. The first simply calculates mean, standard deviation and skewness for each color channel leading to less dimensions in the actual feature vector. Color histograms are slightly different, since they capture the color distribution of an image by quantizing the color space in n bins resulting in a n-dimensional feature vector. Standardization bodies such as the Moving Pictures Expert Group (MPEG)[58] addressed the need for unified multimedia features and proposed a set of features in the MPEG-7 standard [Sik01]: *Dominant color descriptor* is more compact than color histogram and faster to compute. It stores the representative color of the image along with spatial coherence and variance. The *color layout descriptor* has been designed to capture color distribution in an arbitrary-shaped region. Within MPEG-7, the color histogram has been lifted to a more generic representation, namely the *scalable color descriptor*. This histogram is encoded with a wavelet transformation [ABMD92] and uses the HSV color space with a 255 bin quantization.

Texture features. In contrast to color, texture-based information can only be derived from an image while considering a group of pixels. The main aim of such approaches is to find homogeneous areas, such as the sea in the background of an

[58] http://mpeg.chiariglione.org/, last checked December 18, 2013.

image. Those approaches are divided in *spectral* and *spatial texture feature extraction methods*. The first group utilizes algorithms of the frequency domain, such as Fourier transform [LC05] or Gabor filters [JCH04] to extract the desired information. A further consideration of this group is not part of this thesis. Spatial-based extraction of features is mostly based on structure, statistics or a specific model. A *grey level co-occurance matrix* is an example for an statistical approach. In general, these features find textures while processing the inter-pixel distance and the orientation for all pair wise combinations of grey levels in the spatial region [Cla02]. The aforementioned MPEG-7 standard introduces a non-homogenous texture descriptor, the *edge histogram*. This descriptor splits the image in 16 non-overlapping blocks with equal size. After partitioning, for each block one of five edge categories[59] are applied leading to five bins.

Shape features. The occurrence of specific objects inside an image is a relevant information in terms of knowledge representation. A common approach is to detect the shape of an objects in order to perform a detection on this knowledge. In literature, a shape is an equivalence class of geometric objects invariant under translations, rotations and scale changes keeping up the aspect ratio [Rue10]. In terms of shape features, two general categories arise: *contour* or *region-based methods*. In general, region-based algorithms are more robust to noise than contour-based ones, since the latter only take the sphere of pixels around the contour into account. Besides proprietary shape features, the MPEG-7 standard defines features for both categories. The contour-based shape descriptor makes use of curvature scale-space representations, which include spleen and circular information of the detected contour. The region-based descriptor *angular radial transformation* employs moment invariants [TC92], which are per definition invariant to transformation.

Spatial relationship. The information on color distribution as well as occurring objects is very helpful in terms of the multimedia indexing task. If objects have been detected and/or identified, the spatial relation between them can be also exploited by retrieval engines. In general it is distinguished between an *absolute, pixel-based specification* of objects and *relative locations*, such as left, right or above.

Image segmentation. Region-based feature extraction starts with partitioning an image. The resulting fragments serve as an input for the actual feature extraction algorithm leading to a feature representation for each fragment. Optionally, a post-processing step may aggregate the calculated feature representations to a globally valid result. Within image segmentation, the main task is to compute the image segmentation. Current research tries to solve this by the use of clustering algorithms, contour-based segmentation, statistical models or graph based approaches. A survey of segmentation techniques can be found in [PP93].

[59] The edge categories are as follows: vertical, horizontal, 45°, 135°, and non-linear edges.

For a comprehensive consideration of visual features in the domain of image retrieval one is reffered to the evaluation conducted by Deselaers et al. in [DKN08] as well as Tuytelaars et al. [TM08].

4.3 Similarity Measures

The extraction of multimedia features is the first step towards content-based retrieval. As introduced, multimedia features are expressed by n-dimensional feature vectors holding numerical values. At its core, a similarity search calculates all distances between the input feature vector and the feature vectors stored in the database following a specific function, see Figure 2.1. Those functions follow the mathematical definition of a *metric space* and a *metric* [Bry85].

In literature, there exist a vast number of metrics that are grouped into so-called metric families. One of the oldest and most frequently used metric family is the L_p Minkowski family [Cha07]. Here p is a parameter defining the p-th root and the p-th power as shown in Table 4.1. Figure 4.3 illsutrates the difference of the Cityblock L_1 and Euclidean L_2 metric in a graphical way. In general, for each multimedia feature exists a recommended best practice to calculate the distance on. In terms of visual information retrieval, Eidenberger evaluated various similarity measures for their application in the domain of MPEG-7 [Eid03].

Name	Formula
Cityblock L_1	$d_{CB} = \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$
Euclidean L_2	$d_{Euc} = \sqrt{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^2}$
Minkowski L_p	$d_{MK} = \sqrt[p]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^p}$
Chebyshev L_∞	$d_{Cheb} = \max_i \lvert P_i - Q_i \rvert$

Table 4.1: L_p Minkowski Family

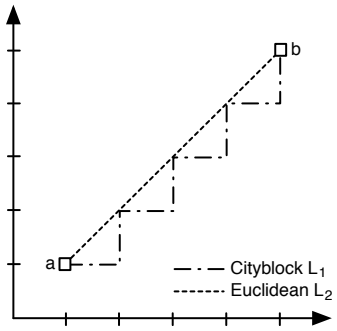


Figure 4.3: Graphical representation of Cityblock L_1 and Euclidean L_2 metric

4.4 Characteristics of Similarity Query Types

The consideration of low-level multimedia features and similarity measures leads to the last building block of content-based retrieval, namely the realization of the similarity search itself. In literature three basic types of similarity queries are distinguished [AFS93, Sei98]:

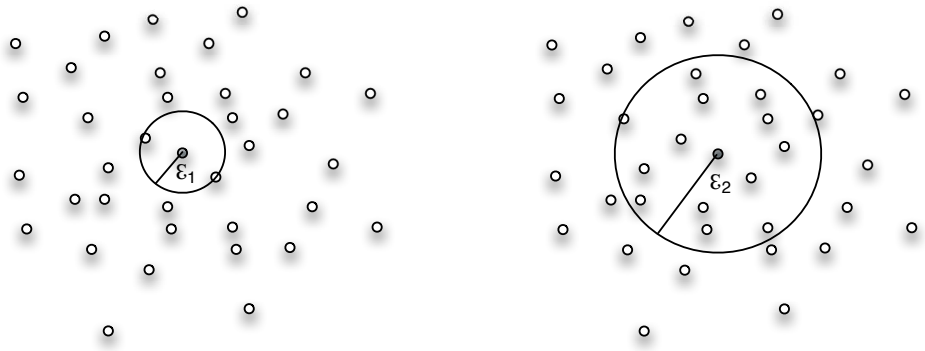
Definition 1 (*Similarity Range Query (ε-similarity)*)

Let O be the set of all query objects, q be a query object $q \in O$ and ε be a

query range $\varepsilon \in \mathbb{R}$. *The Similarity Range Query returns the set:*

$$sim_q(\varepsilon) = \{o \in DB | d(o,q) \leq \varepsilon\}, \text{ with}$$

o be a multimedia object stored in the database DB and d be a similarity measure.



(a) few results (b) many results

Figure 4.4: Correlation between ε and result set size in Similarity Range Query

The definition of ε-similarity clearly shows, that the result set size is unknown during query execution. Multimedia objects are being selected as elements of the result set, if the distance is less or equal to the threshold ε. This correlation is illustrated in Figure 4.4.

The imprecision in terms of the result set size present in ε-similarity can be a major drawback in certain use cases leading to a computational overhead in the worst case. The following query type addresses exactly this issue:

Definition 2 (*Nearest Neighbor Query*)

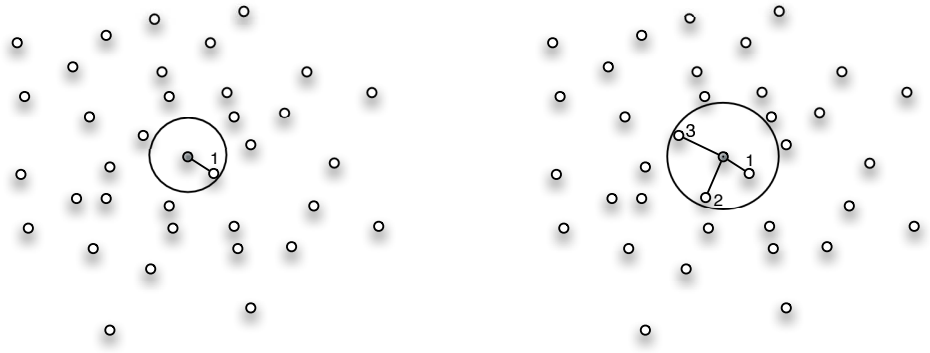
Let O be the set of all query objects and q be a query object $q \in O$. *The Nearest Neighbor Query returns the set* $NN_q \subseteq DB$:

$$NN_q = \{o \in DB, \forall o' \in DB | d(o,q) \leq d(o',q)\}, \text{ with}$$

o be a multimedia object stored in the database DB and d be a similarity measure.

The Nearest Neighbor Query returns the closest multimedia object for the query as illustrated in Figure 4.5 (a). If there exist more multimedia objects with the same minimal distance, all of them are returned. If two objects exhibit a distance equal to 0, they are considered to be equivalent.

In most cases, the Nearest Neighbor Query is too restrictive in terms of finding only the most similar object. Due to this fact, the well-known *k-Nearest Neighbor Query*, has been introduced:



(a) Nearest Neighbor Query (b) k-NN ($k = 3$)

Figure 4.5: Examples for different assignments of k in Nearest Neighbor Query

Definition 3 (*k-Nearest Neighbor Query (k-NN)*)

Let O be the set of all query objects, q be a query object $q \in O$ and k a query parameter. The k-Nearest Neighbor Query returns the set $NN_q(k) \subseteq DB$ containing (at least) k multimedia objects:

$$\forall o \in NN_q(k), \forall o' \in (DB - NN_q(k)) : d(o, q) < d(o', q), \text{ with}$$

o be a multimedia object stored in the database DB and d be a similarity measure.

In this regard, k-NN is a more generic version of the Nearest Neighbor Query, where k specifies the maximal amount of multimedia objects in the result set, see Figure 4.5 (b). Furthermore, Figure 4.5 (a) can be also noted as k-NN with $k = 1$.

4.5 Accessing Multimedia Features

With the prior introduced components, a complete content-based retrieval system can be established. However, efficient retrieval in large sets of multimedia objects is not possible due to the *sequential scan* of the input multimedia feature with all multimedia features stored in the database. In the multimedia domain, well-known algorithms of the database community have been adopted and extended to enable hierarchical partitioning of the extracted multimedia features. This process is termed *multimedia indexing* [DGB06].

In traditional database systems, tree-based data structures such as B-Tree [Bay72] and its variations are in use. Due to their one-dimensional characteristics, those data structures are insufficient for the multimedia domain. A reasonable high-dimensional index structures has to consider at least the following requirements [GG98]:

- *Correctness & completeness* must be ensured in terms of the result set. The result set must stay the same in size and ranking as without index structures.

Table 4.2: Distinction of tree-based index structures

Type	Distinction
Segmentation	space segmentation, data segmentation
Segment separation	overlapping, disjunctive
Composition	balanced, unbalanced
Storage	leaves and nodes, leaves only

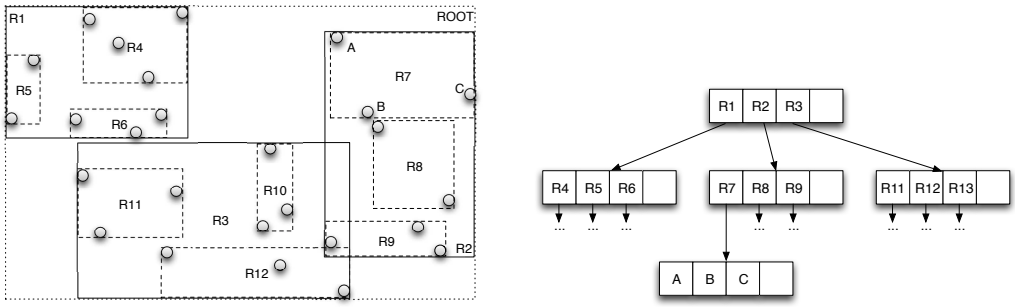
- *Scalability* guarantees a constant efficiency with an increasing amount of dimensions.
- *Search efficiency* decreases the amount of executed data access calls in comparison to the sequential scan.
- *Support of query types* such as introduced in Section 4.4 to be compliant to a broad range of use cases.
- *Support of CRUD operations*[60] for a sophisticated management of the content of a index structure.

In addition to the set of requirements, a high-dimensional index structure exposes diverse characteristics as summarized in Table 4.2 [BBK01]. The first characteristic is the way, a index is performing its segmentation. Here, the division is made in approaches dividing the overall data space (e.g., Segment Tree [BW80]) or the actual data points (e.g., B-Tree). The separation itself is the second point, whether the created segments overlap or are disjunctive in terms of set theory. The last two characteristics deal with the overall structure of the index. There exist tree-based index structures which are balanced in means of height or filling degree, such as AVL-Tree [AVL62]. In contrast to that, unbalanced tree-based exist, such as the binary search tree [Bla11]. Finally, the storage of the actual data can be done in both, inner-nodes and leaves (e.g., T-Tree [LC86]), or in leaves only (e.g., B-Tree).

Since 1960, the research community designed a plethora of multidimensional access methods. An extensive summary has been proposed by Gaede and Günther [GG98] highlighting the theoretical background as well as the evolution of this research field. In terms of multimedia retrieval, the *R-Tree family* are heavily in use and will be discussed next.

Guttmann issued the classic version of the R-Tree [Gut84] in 1984 with the aim to index spatial data. At its core, it can be seen as an extension of the B-Tree index while considering multiple dimensions. The main idea is to partition the

[60] In the database domain, CRUD operations are the atomic operations in a database system: Create, read, update, and delete.



(a) Graphical representation (b) Tree-based representation

Figure 4.6: Example for a R-Tree index structure

data space in *minimal bounding rectangles* (MBR) as shown in Figure 4.6 (a). In a R-Tree, a pre-configuration defines the minimal (m) and maximum number (M) of entries for a MBR, with $m \leq \frac{M}{2}$. In the given example, $m = 2$ and $M = 4$ is chosen. Each MBR must enclose every all MBRs in the lower level.Similar to the B-Tree, the data is stored in the leafs, whereas inner nodes represent the MBRs, see Figure 4.6 (b). The R-Tree is height-balanced, however, MBRs may overlap. Obviously, the degree of MBR overlaps leads to an increase of the overall search complexity due to multi-lookups in the rectangle regions.

To soften the negative effect of overlapping MBRs in the R-Tree, Sellis et al. introduced the R$^+$-Tree [SRF87]. The R$^+$-Tree minimizes the *coverage* of an MBR as well as the amount of *overlaps* by allowing the following three central conditions: (i) nodes may be filled with less elements than guaranteed by m, (ii) inner nodes do not exhibit an overlap and (iii) an object may be stored in more than one leaf node. On the one hand, the advantage of this index structure is the improvement of the point query performance, but on the other hand it might get larger than an R-Tree due to duplicates leading also to a more complex maintenance.

Beside coverage and overlap, the overall structure of the R-Tree is highly dependent on the insert sequence of the objects in terms of node splitting. Beckmann et al. took this observation into account while creating the R*-Tree [BKSS90]. In the original version of the R-Tree, only the minimal coverage of a MBR was taken into account. The R*-Tree utilizes forced reinsertion of elements along with the following constraints while its insert and splitting methods: (i) area covered by a MBR should be minimized, (ii) overlaps should be minimized, (iii) sum of edge lengths should be minimized and (iv) optimize storage utilization. The last constraints leads to a smaller height of the index structure and guarantees efficient point queries[61].

The SS-Tree [WJ96] is intended to improve the performance of nearest neighbor queries of the R*-Tree. At its core, it uses spheres instead of rectangles and adapts the insert and split methods to the specific needs of this query type. The center of

[61] A point query is defined by Beckmann et al. as follows: Given a Point P, find all rectangles R in the tree with $P \in R$.

a sphere is the centroid of the enclosed objects. The SR-Tree [KS97] proposed by Katayama and Satoh combines the usage of rectangles and spheres. Here, a region is described by the intersection of a bounding sphere and a bounding rectangle improving nearest neighbor queries for high-dimensions and non-uniform data.

4.6 The Curse of Dimensionality and Beyond

As the prior chapter shows, there exist lots of variants from the classical R-Tree. Nearly all of them expose various advantages for certain query types and domains. Their multi-dimensional structure reduces the actual data access hits to perform efficient similarity search over the whole data corpus[62]. However, due to their structure basic operations such as insert or delete can be very cost intensive. Experiments [BBK98] have shown that multi-dimensional index structures ensure efficient access until an approximate number of 20 dimensions have been reached. From there on, the sequential scan over the whole data corpus is faster than using the multi-dimensional access method. This phenomenon is called *curse of dimensionality*.

A recent analysis [Sam10] by Samet dedicated to the curse of dimensionality issue of multimedia indexing with a focus on nearest neighbor and range queries. One of his conclusions is that *promising research directions lie in developing techniques to identify the important features in the applications so that the dimension of the problem domain can be reduced*. Following this, there have been efforts to estimate the ability of multimedia features to index large corpora [BGBR+10] by cross-evaluating various features along with a set of common index structures. Further, techniques have been developed to significantly reduce the amount dimensions: Huang et al. [HSS+08] proposed an approach to reduce the dimensions in top-k image retrieval by locality condensation. In this light, locality condensation means that the locality information of the image neighborhood will be combined with the global similarity without producing overlaps in the extracted low-level information of the image segment. Similar to this, Chen et al. [CQL11] introduced a nonlinear adaptive dimension reduction algorithm. It adaptively learns the ideal dimensions storing the specific geometric information of the image. Machine learning techniques are also heavily in use to reduce the amount of dimensions: Urrutu et al. [UDJ08], Shen et al. [SOZ05] and Huang et al. [HSLZ11] employ clustering techniques, whereas support vector machines [RM12] have been also considered. Besides, map reduce [YFM+09, WYLD10, MSGA13] are also adopted to manage the amount of dimensions. By the help of map reduce, the multimedia community tries to process large multimedia corpora by parallel and distributed operating algorithms in huge computing clusters.

Despite reduction of dimensions, research efforts additionally focused on the index structures themself: Valle et al. [VCPF08] introduced multicurves to index

[62] Here the term universe describes a set of media resources on which a retrieval process is performed.

the dimensions of multimedia features. The novel approach is to use a set of space filling curves, e.g., Hilbert curves [Sag94], in which every curve is responsible for a specific subset of the dimensions. This approach improves the processing of k-NN queries as well as the management of the data structure. Similar to multicurves, the recently proposed NV-Tree [LJA11] has been designed to efficiently solve k-NN queries, too. At its core, it stores each high-dimensional feature in a 6-bit long array. From a structural point of view it combines combination of projections of data points to lines and partitioning of the projected space. In contrast to that, well-known index structures have been also refined. Beckmann et al. [BS09] redesigned the R*-Tree to be fully compliant to a relational database management system. This has been achieved by reengineering the subtree selection as well as split algorithm to ensure a single path tree. By following such integration approaches into relational database systems, researchers try to create an efficient native support for querying multimedia documents.

Especially in the Web domain, the multimedia research community recognized the need to aggregate information of different features that are exposed by various expert systems or attached to the media resource. This topic is called *feature fusion* and is heavily used to soften the semantic gap to improve retrieval capabilities [PG08, CTZZ10] as well as summarization of media resources [DMR+12].

In this thesis, syntactic multimedia features are applied within the *late fuzzy multimedia fusion* approach. Here, they create on-the-fly content-dependent abstractions of media resources to enable an aggregation of unfederated retrieval services. Section 4.2 already introduced several multimedia features that can be seen as well-accepted by the multimedia community. Those will be further discussed and evaluated in Chapter 11 whether they can be applied to the fusion technique in an adequate way.

Chapter 5

Multimedia Retrieval Systems

Traditional relational databases are not sufficient to integrate the aforementioned components to enable efficient multimedia retrieval out of the box. Insufficient support of similarity search, multidimensional index structures as well as exploitation of multimedia semantics while querying are only a few limitations. Architectures for multimedia information retrieval face those new frontiers by proposing solutions stemming from all disciplines of computer science, such as media processing, signal processing, data mining or database technologies [DSL05].

This chapter focuses on retrieval technologies and gives insights into *multimedia retrieval systems*. In particular, Section 5.1 defines terminology and requirements for multimedia retrieval systems. Depending on the intended usage domain, specific architectures can be chosen as illustrated in Section 5.2. The chapter is completed by the consideration of multimedia query languages (cp., Section 5.3) as well as query processing in multimedia retrieval systems (cp., Section 8.7).

5.1 Terminology & Requirements Definition

Early research efforts already observed the need for specific systems capable to manage multimedia enriched user requests. In this regard, the term *multimedia database management system* has been informal introduced by Christodoulakis at the SIGMOD conference series in 1985 as follows [Chr85]:

Term 20 (*Multimedia database management system*)

> *The term multimedia database management system refers to the problems of managing unformatted data, as well as to the problems introduced by the devices, which are used for the presentation and storage of unformatted data.*

In addition to Term 20, a catalog of nine problems was defined that were meant to be essential for the creation of multimedia database management system1:

i) Software architecture defines the characteristics of the overall system. Systems can be designed standalone or as an extension of another system focusing on specific or rather generic application domains.

ii) Content addressability means how the content of a media resource can be accessed, e.g., a person in a video shot.

iii) Performance has to be ensured by exploration of index structures, clustering techniques as well as high-end hardware architectures.

iv) User interfaces shall be able to cope with the presentation of diverse media resources. An crucial point is the establishment of an active communication on the basis of user interactions.

v) Information extraction, transformation and correlation especially from very large archives or digital libraries has to be supported. Further, new knowledge could be derived from the extracted concepts.

vi) Concurrency control, recovery, security and version support shall be investigated and supported by this systems.

vii) Large capacity storage devices must be available to store especially the raw or uncompressed media resources.

viii) Information retrieval techniques such as similarity search shall be integrated in this systems.

ix) Working prototypes ensure evolvement of the systems by ongoing user evaluations.

Within the last three decades, several issues of this catalogue have been softened. Advancements in the area of persistent storage, e.g., hard-disks and cloud computing, lessened issues regarding performance and storage. Further, extension of relational databases regarding new data types made it possible to manage media resources inside databases. As already mentioned, multidimensional index structures are currently well understood in terms of their limitations and various techniques have been proposed. The outcome of those activities led to a plethora of multimedia retrieval systems produced by research activities, e.g., PythiaSearch [ZBB+12], or industrial efforts, e.g., IBM multimedia search and retrieval system [NTX+07]. Up to now, the definition of Christodoulakis still holds and coins the current understanding of the multimedia communities rather informal definition of a *multimedia information retrieval system*. Lew et al. [LSDJ06] specify a multimedia information retrieval systems by defining two fundamental requirements: *retrieval* of relevant documents and *browsing* a multimedia collection. Regardless, several other issues needs to be addressed in future research to reach the requirements. As one will see next, especially in the Web domain, software architectures (c.p., Section 5.2) plays an important role since this environment is highly dynamic and versatile. In addition, attempts have been made to define new query languages (c.p., Section 5.3) integrating content-based similarity search techniques in traditional databases.

5.2 Architectural Facets

The current trend of multimedia information retrieval in the Web goes towards distributed retrieval services. This follows the overall nature of the Web itself. In terms of Social Media, a vast amount of blogs (152M.) and social networks with millions of user profiles (approx. 175M. accounts on Twitter and 600M. on

Facebook) serve as entry points to share multimedia resources in an easy fashioned way. Within those services, several billions of user-generated multimedia resources are publicly retrievable on Social Media sharing platforms such as Flickr, Picasa or YouTube through APIs. Since this thesis is concerned about the distributed character of multimedia retrieval, this section presents fundamental architectural characteristics to create multi-database systems [SL90]. Further, their relevance in the domain of multimedia retrieval are outlined. Figure 5.1 shows the relation between the different architectures, which will be discussed in the subsequently. A detailed consideration of distributed relational database systems can be found in [Rah94].

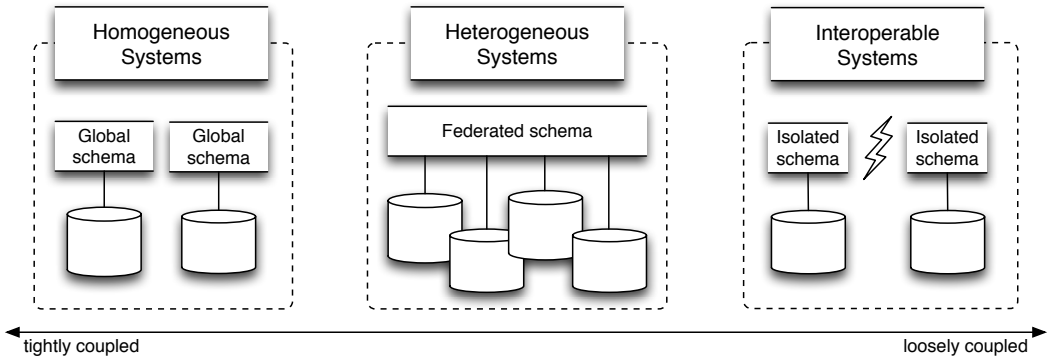


Figure 5.1: Overview of architectures for distributed multimedia retrieval

Homogeneous systems are distributed retrieval environments that exhibit a tight coupling between the physically detached database systems. The basic idea behind this architecture is that all nodes are uniform regarding the global schema and the query language or API. In contrast to fully replicated databases, the stored data must not overlap in homogeneous systems. On the one hand tight coupling enables simplified retrieval processes, but on the other hand complicates maintenance.

Heterogeneous systems constitute of a set of single databases where each of them has been independently designed and implemented. *Heterogeneity* in such an environment is present at the schema level, the constraints defined on the schema as well as the used query language. Further, due to diverse schemas, semantic heterogeneity may occur while interpreting the data. Nevertheless, heterogeneous systems tend to operate on (various) federated schemas in order to exclude semantic heterogeneity. Obviously, the connected databases must ensure a certain degree of autonomy as well as willingness to cooperate in the creation of federated schema(s) and a global query execution. Sheth and Larson [SL90] differentiate between *design*, *execution* and *association autonomy*. Design autonomy often hinders federated schemas. In contrast to that, execution (local execution of without interference) and association autonomy (allocation of functionality and resource) promote federation.

The benefit of federation lies in hiding the diversity of structure, location and naming conventions used in the data bases.

Interoperable systems often follow the conceptual design of *mediator systems* stemming from the well-known *broker* architecture [Gom11]. Here, several databases are loosely coupled by the mediator. This approach has been termed *result fusion* or *metasearch* in the Web Information Retrieval community, denoting techniques that combine pre-ranked results from multiple search engines into one consistent result. Montague and Aslam differentiate between two different types of metasearch techniques: *internal* and *external metasearch* [MA02]. External metasearch treats existing search engines which are potentially operating on diverse document sets as black boxes and consolidates their output. Internal metasearch engines combine multiple sub-engines that are operating over the same set of documents. Important phases in result fusion include, depending on the applied strategy, the normalization of scores, the elimination of duplicates, the re-ranking of results, and the aggregation of the result lists.

Recapitulatory, there exist a huge amount of multimedia retrieval services on the Web; their varying usage scenarios lead to the application of various standardized or proprietary multimedia metadata schemas, c.p., Table 3.1 of Section 3. It is quiet obvious that homogeneous architectures are not able to cope with this diversity and are in general not suitable to be applied to distributed multimedia retrieval. In contrast to that, heterogeneous and interoperable systems fit the requirements better. Whenever a federated schema can be constructed, the heterogeneous systems should be chosen. Due to cooperative sub-systems, a global query execution plan can be applied and potentially optimized, e.g., with aligned techniques stemming from relational systems [Cha98]. Without this assumption, interoperable systems are the only way to establish a distributed multimedia retrieval environment. Here, the mediator loosely connects the retrieval services and undertakes tasks such as schema alignment, query transformation or result aggregation. In both, heterogeneous and interoperable architectures, a single query language or API has to be selected as abstraction layer from the underlying retrieval paradigms.

5.3 Multimedia Query Languages

The last building block of a full fledged (distributed) multimedia database management system is the decision of an appropriate query language. The vision to design a multimedia query language, that enables an unified access to multimedia data, resulted in many approaches. Within the last decade, various proposals have been issued: extensions of SQL (e.g., SQL/MM [ME01]) and Object Query Language (OQL) (e.g., POQL^{MM} [LCH01]), languages bound to a specific metadata model [CMP02], languages concentrating on a special retrieval technique (e.g., TVQL [HR96] for temporal retrieval) or languages that integrate weighting capa-

bilities for expressing user preferences (e.g., WS-QBE [SSH05]). Since many multimedia metadata schemas rely on representational models (c.p., Section 3.3.1) and multimedia ontologies (c.p., Section 3.3.2) query languages such as XQuery [Cha02] and SPARQL [PS08] are also in use for multimedia retrieval systems. In this thesis, the MPEG Query Format (MPQF) [DTG+08] has been considered to be used due to its recent publication as international standard, its full-fledged support of multimedia queries and d its domain independence. It specifies a format for the interaction of multimedia clients and multimedia retrieval systems serving as an abstraction layer between clients and the underlying retrieval paradigms. In detail, the standard defines the message format for multimedia requests (e.g., query by example or metadata-based querying) to heterogeneous multimedia retrieval systems and the message format for their responses. Furthermore, a management part provides features such as service discovery and service capability description. The features of MPQF will be presented in more depth in Chapter 8.4.

The origin of fast processing of query requests lies in a formal model of the underlying query language and its optimization capabilities, namely in an associated query language algebra. The relational algebra defined by Codd [Cod70] is a recognized substitute for the relational database domain, which specifies the language constructs and their behavior. A lot of research has been already done in defining algebras for multimedia query languages: Atnafu et al. [ABK01] formally defined similarity-based operators (e.g., similarity-based range query) and aligned them with operators available in the relational algebra. A similar approach has been issued by Montesi et al. [MTD03], Schmitt and Schulz [SS04b]. To enable similarity search, they enlarged the relational algebra by fuzzy algebra offering new operations and the possibility to include weighting factors representing user requirements. Wu et al. [WLLC10] also enlarged the relational algebra to build a formal basement for the multimedia query language UMQL by extending specific algebraic operators. Döller et al. [DLKS11] also proposed an algebra for MPQF by reducing its structure to principles of quantum logics [Sch08]. In terms of XML query languages, two main possibilities in defining an algebra can be distinguished: tuple- and tree-based approaches. Natix algebra [BHKM05] or BSA algebra [SBH06] belong to the family of tuple-based approaches relying mappings from XML instances to relational tuples and XML-related extensions of the relational algebra. Algebra of the second approach focus on a tree-based representation (e.g., TAX [JLST01]) for optimization tasks. Furthermore, algebras have been issued that enable algebra-based query refinement on the basis of Semantic Web ontologies [ZW04] to soften the semantic heterogeneity between various models.

Part III

Improving Metadata Interoperability

Chapter 6

Unified Access to Multimedia Metadata

This chapter focuses on the first contribution of this thesis, namely the contributions to the *improvement of the metadata interoperability issues* by establishing an community-driven, international standard to create and retrieve uniform media resource annotations[63]. This approach softens interoperability issues present at the modeling (M1) and instance level (M0), which have been introduced in Figure 3.2 of Section 3.3. The chapter is structured as follows: Section 6.1 introduces related work done in this domain. A consideration of use cases and requirements in Section 6.2 clearly shows the benefits of the pivot metadata format for media resources, highlighted in Section 6.3.

6.1 Related Work

The overview in Table 3.1 showed that many metadata schemas have been created to improve the interoperability between different systems within one domain or application type. In this section[64], well-known image and video metadata schemas will be introduced and approaches for combining them are discussed. An exhaustive list of multimedia metadata schemas currently in use has been produced by the W3C Multimedia Semantics Incubator Group[65]. This list has been taken into account for the following consideration.

6.1.1 Many Standards for Different Needs

Looking into the domain of still images several popular metadata schemas can be identified: Photos taken by digital cameras are annotated with Exchangeable Image File (EXIF[66]) metadata directly embedded into the header of image files. It provides technical characteristics such as the shutter speed or aperture, and contextual information (date and time) of the captured image. The Extensible Metadata Platform (XMP[67]) is a specification published by Adobe for attaching metadata to

[63] This Chapter is partially based on [SBB+13].
[64] This Section is partially based on [SBB+09].
[65] http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/, last checked December 18, 2013.
[66] http://www.digicamsoft.com/exif22/exif22/html/exif22_1.htm, last checked December 18, 2013.
[67] http://www.adobe.com/devnet/xmp/

media assets in order to enable a better management of multimedia content. The specification standardizes the definition, creation, and processing of metadata by providing a data model, a storage model, and formal predefined sets of metadata property definitions. XMP makes use of RDF in order to represent the metadata properties associated with a document. The DIG35[68] specification of the International Imaging Industry Association (I3A) defines a standard set of metadata for digital images including basic image parameter, image creation (àla EXIF), content creation and intellectual property rights and represented in XML. The IPTC Photo Metadata standard[69] developed by the International Press Telecommunication Council (IPTC) provides also a set of metadata properties being administrative, descriptive or related to the image rights. Largely based on XMP, this specification allows to represent as well complex semantic descriptions of the subject matter (e.g. persons, organizations, events).

In terms of video EBUCore[70] is an XML-based metadata standard created by the European Broadcasting Union (EBU) consisting in a set of metadata properties specializing Dublin Core for describing radio and television content. The already introduced MPEG-7 standard issued by the Motion Pictures Expert Group (MPEG) creates comprehensive and domain independent description of audio, video and multimedia content. It is especially designed for document retrieval. The standard is based on XML Schema but MPEG-7 ontologies expressed in OWL have been proposed as outlined in Section 3.5. The standard is composed of many descriptor tools for diverse types of annotations on different semantic levels, ranging from very low-level features, such as visual (e.g. texture, camera motion) or audio (e.g. melody), to more abstract descriptions. The flexibility of MPEG-7 is based on structuring tools, which allow descriptions to be associated with arbitrary multimedia segments or regions, at any level of granularity, using different levels of abstraction.

This excerpt clearly shows that numerous metadata standards exist for annotating multimedia resources, all with their own benefits and community usage. It is undesirable to create a single multimedia metadata schema that would satisfy all use cases. Thus considerable efforts have been made to lift non Semantic Web-aware formats, e.g., MPEG-7 into RDF to soften the interoperability issues on the M2 level, critical interoperability issues in levels M1 and M0 still remain. Some additional steps are needed to combine these formats and interoperability can be achieved by the means of mappings or relationships between the different schemas.

6.1.2 Interoperability Approaches between Metadata Schemas

Xing et al. [XXE07] present a system for automatic transformation of XML documents using a tree matching approach. However, this method has an important restriction: the leaf text in the different documents has to be exactly identical. This

[68] http://xml.coverpages.org/FU-Berlin-DIG35-v10-Sept00.pdf

[69] http://www.iptc.org/std/photometadata/2008/specification/IPTC-PhotoMetadata-2008_2.pdf

[70] http://tech.ebu.ch/docs/tech/tech3293-2008.pdf

is hardly the case when combining different metadata standards. Likewise, Yang et al. [YLL+03] propose to integrate XML Schemas. They use a more semantic approach, using the ORA-SS data model to represent the information available in the XML Schemas and to provide mappings between the different documents. The ORA-SS data model allows to define objects and attributes to represent hierarchical data, however more advanced mappings involving semantic relationships cannot be represented.

Cruz et al. [CXH04] introduced an ontology-based framework for XML semantic integration. For each XML source integrated, a local RDFS ontology is created and merged in a global ontology. During this mapping, a table is created that is further used to translate queries over the RDF data of the global ontology to queries over the XML original sources. The authors assume that every concept in the local ontologies is mapped to a concept in the global ontology. This assumption can be hard to maintain when the number and the degree of complexity of the incorporated ontologies increases. Poppe et al. [PMMdW09] advocates a similar approach to deal with interoperability problems in content management systems. An OWL upper ontology is created and the different XML-based metadata schemas are represented as OWL ontologies and mapped to the upper ontology using OWL constructs and rules. However, the upper ontology is dedicated to content management system and, as such, is not as general as the approach proposed in this thesis.

Other standardization bodies also recognized the need for metadata interoperability: JPSearch [DAE07] is a project issued by the JPEG standardization committee to develop technologies that enable search and retrieval capabilities among image archives, consisting of five parts. While the first part focus on describing use cases and the overall architecture of image retrieval systems, the part 2 introduces an XML-based core metadata schema and transformation rules for mapping descriptive information (e.g., core metadata to MPEG-7 or core metadata to Dublin Core) between peers [DSK+10]. Part 3 adapts a profile of the MPEG Query Format for ensuring standardized querying. Part 4 adopts the well known image data formats (JPEG and JPEG 2000) for embedding metadata information. The benefit of such an integration and combination of metadata with raw data is the mobility of metadata and its persistent association with the image itself. By embedding the metadata into the image raw data file format, one improves the flexibility within the annotation life cycle. However, the interchange of image data between JPSearch compliant systems remains an open issue. For this purpose, Part 5 concentrates on the standardization of a format for the exchange of image or image collections and its metadata and metadata schema between JPSearch compliant systems.

6.2 Use Case & Requirements

The main characteristics and thus the essential feature of the Web is its decentralized model of content publishing. This favors current Web trends, such as the already introduced Social Media movement. In this domain, three main parties can

be distinguished: *content provider*, *retrieval service* and *consumer*. This observation is directly applicable to one of the real-world application scenarios of Section 1.1, namely the *isolated image retrieval* use case considering several autonomous image retrieval services offering metadata- and / or content-based retrieval services. Within those, the aforementioned peers interact with each other. A possible workflow here is as follows: a content provider shares media resources on several retrieval services and exposes related metadata information, e.g., title and keywords. In addition to indexing and storage of the information, a portal service potentially infers more metadata. An example for such a metadata enrichments is the lookup of a locations name on the basis of the GPS coordinates stored in the given metadata or media resource. In general, the consumer is actively searching for the media resource or is attracted to it while browsing the retrieval service.

To analyze this situation a little bit more concrete, the query added to the isolated image retrieval use case in Section 1.1 will be used:

> *"Give me the first ten images that are similar to http://any.uri/-strawberry.jpg or are annotated with the keyword strawberry!"*

While executing this query in a decentralized and distributed manner, several metadata interoperability issues are present. First, a content provider may utilizes different metadata schemas for annotating the actual media resource (in this example an image), e.g., EXIF for technical description of the images such as shutter speed or GPS location whereas Dublin Core for covering descriptive annotations such as title and keywords. Second, there exist a large number of image retrieval services for image sharing like Flickr or Photobucket[71]. All of them use their own internal (proprietary) metadata scheme to structure metadata information. In this context, we consider a mediator system that enables user to consume a media resource by abstracting from the actual distributed retrieval environment. Obviously, a harmonization of the diverse metadata schemas is essential for this task. To solve the given query, an alignment has to be present to distribute the metadata-based information to the appropriate retrieval services in such a way that the metadata information can be further processed internally.

Following the real-world application scenario, a set of requirements for a pivot metadata schema as well as an API for harmonizing the access to media resources can be derived:

- *Composition:* The design of the ontology and API provides support for structured metadata and controlled vocabularies wherever possible, but do not enforce their use.

- *Coverage:* The ontology and the API are not bound to a specific application domain, media type or content representation.

[71] `http://photobucket.com/`, last checked December 18, 2013.

- *Extensibility:* Due to the flexible structure of the Web, future versions of the specification may contain additional properties in the core vocabulary (and its representations) and mappings to more metadata schemas.
- *Granularity:* The ontology and the API can be used independently, depending on the actual application domain. Further, conformance to the specifications is possible on different levels of strictness.
- *Interoperability:* Syntactic and semantic interoperability is ensured by the defined semantics of the set of core properties and the mapping tables to the metadata schemas in scope.

Besides the application in the isolated image retrieval use case, the proposed metadata scheme is also in use within the federated medical retrieval use case due to its domain independence.

6.3 A Pivot Metadata Scheme for Media Resources

The work on a community-driven, standardized pivot metadata scheme for multimedia resources has been initiated by the W3C by launching the Multimedia Semantics Incubator Group[72] in 2007 with the goals to analyze the metadata interoperability issue for multimedia on the Web and to show the feasibility of using Semantic Web technologies to align different multimedia metadata formats. The outcome of this group led to the foundation of the W3C Video on the Web activity, which amongst others hosts the Media Annotation Working Group[73] aiming to improve the interoperability between multimedia metadata formats. Within this working group, i contributed to essential stages of the standardization process as shown next:

The main output of the Media Annotation Working Group is the *Ontology for Media Resource 1.0* [LBB+12]. The purpose of the ontology is to overcome the current proliferation of multimedia metadata formats by providing mappings from properties in different formats to a common set of properties in the ontology. The ontology is accompanied by the *API for Media Resource 1.0* [SBH+13] that provides uniform access to the elements defined by it. Within this process i was determinative of the mapping tables composition and validation. Besides, i am the main editor of the API specification and therefore heavily influenced its overall design. Further, i primary developed central implementation prototypes of both specifications. Those were needed to move both specifications to the status of an official W3C recommendation and to . From a dissemination point of view, i was the main driver as well as main author for almost all papers written in the context of the working group.

For both, conformance to the specifications is possible on different levels of strictness, from a basic support of the set of properties as key/value pairs up to

[72] http://www.w3.org/2005/Incubator/mmsem/, last checked December 18, 2013.
[73] http://www.w3.org/2008/WebVideo/Annotations/, last checked December 18, 2013.

a formal compliance to the RDF/OWL ontology as well as the API specification. The ontology and API provide support for structured metadata and controlled vocabularies wherever possible, but do not enforce their use. Furthermore, the ontology and API can be used independently, e.g., in a Linked Data use case, the OWL ontology could be used alone, while a Web application might integrate the API only.

6.3.1 Ontology for Media Resources 1.0

The set of core properties defined in the Ontology for Media Resource 1.0 consists of 20 descriptive (i.e., identifiers, language, contributors, creation date, genre, rating etc.) and eight technical (i.e., frame size, duration, format) metadata properties. The descriptive properties are media agnostic and also apply to descriptions of multimedia works (e.g., a movie) that are not specific to an instantiation (e.g., an AVI file). The technical properties, bound to certain media types, are only essential when describing a particular instantiation of the content. Following the requirements of a core vocabulary, all properties are defined with explicit semantics to clarify and disambiguate their definitions in the context of a media resource description. Whenever these properties exist in other standards the Ontology for Media Resource 1.0 explicitly defines how they are related. Furthermore, the ontology can be used with different layers of conformance. If an extension of the basic property semantics is needed, optional subtypes can be used to further qualify many of the descriptive properties, e.g., to define a specific kind of contributor. A detailed overview of the properties can be found in Tables B.1 to B.8 of Appendix B.1.

Apart from the description of the properties as key value pairs, an full Semantic Web compatible representation of the Ontology for Media Resource 1.0 based on the W3C recommendations RDF and OWL has been created[74]. As basement, the EBU CCDM5 (Class Conceptual Data Model) for distribution has been chosen and it defines a set of media- and non-media-specific classes. Figure 6.1 illustrates an excerpt of the class model of the Ontology for Media Resource 1.0. The conceptual model, the implementation, and the usage of the Ontology for Media Resource 1.0 are in detail described in [EB11].

6.3.2 Alignment of Metadata Formats

The set of properties modeled in the ontology has correspondences with existing metadata formats currently describing media resources published on the Web. These correspondences have been defined in the form of mappings, with the aim to provide an interoperable set of metadata, thereby enabling different applications to share and reuse these metadata. Specifically, 19 media metadata formats and seveb media container formats have been selected, as listed in Table 6.1. This list of formats is not closed, nor does it pretend to be exhaustive. A future version of

[74]The ontology is available in XML/RDF at: `http://www.w3.org/ns/ma-ont.rdf`, last checked December 18, 2013.

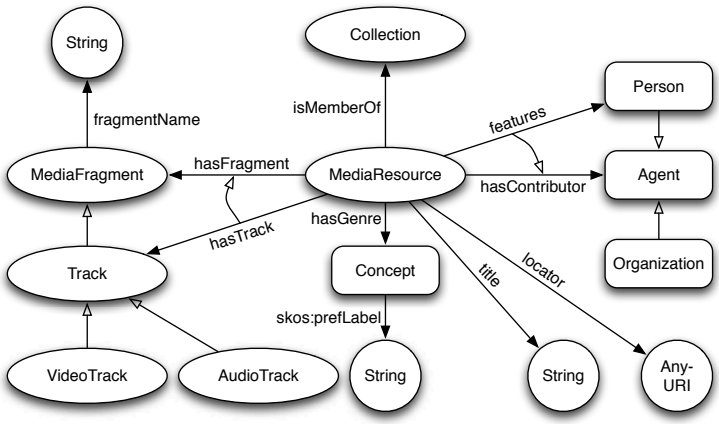


Figure 6.1: Excerpt of the Ontology for Media Resources

Table 6.1: Metadata and container formats considered for alignment

Metadata formats	CableLabs 1.1, DIG, Dublin Core, EBUCore, EXIF, ID3, IPTC, LOM, MediaRDF, MediaRSS, METS, MPEG7, OGG, QuickTime, SMTPD, TVA, TXFeed, XMP, and YouTube
Container formats	3GP, Flash/FLV, Flash/F4V, MPEG4 (MP4), MOV (Quicktime), OGG, and WebM

the Ontology for Media Resource 1.0 may include additional mappings if a need or use case is established for these new mappings. In this light, additional mappings[75] for Schema.org[76] and Atom[77] have been issued as amendment.

The mappings that have been taken into account may have different semantic relations. To express this, the following subset of SKOS[78] are in use to describe semantic relations:

- *Exact match:* Two properties have an equivalent semantics in all possible contexts. For example, the semantics of the Ontology for Media Resource 1.0 property *title* exactly matches the semantics of *CreationInformation/-Creation/Title* defined in MPEG-7.

- *More specific:* The property of the metadata format has a seman-

[75] http://www.w3.org/2008/WebVideo/Annotations/drafts/ontology10/additional-mappings.html, last checked December 18, 2013.
[76] http://schema.org/, last checked December 18, 2013.
[77] http://tools.ietf.org/html/rfc4287, last checked December 18, 2013.
[78] http://www.w3.org/2004/02/skos/, last checked December 18, 2013.

tic that covers only a subset of the semantics expressed by the property defined in the Ontology for Media Resource 1.0. For example, *MediaInformation/MediaProfile/MediaFormat/VisualCoding/Format* or *MediaInformation/MediaProfile/MediaFormat/AudioCoding/Format* defined in MPEG-7 are more specific than the property *format*, because the required use of a classification scheme.

- *More general:* The property of the metadata format has a semantic that covers a superset of the semantics expressed by the property defined in the Ontology for Media Resource 1.0. For example, *SegmentCollection/SegmentRef* defined in MPEG-7 is more general than the *namedFragments* property.
- *Related:* The two properties are related in a way that is relevant for some use cases, but such a relation has no defined semantics. For example, a connection between the *rights* property defined in Dublin Core can be established to *copyright* property.

The one-way mappings defined between the properties of the Ontology for Media Resource 1.0 and the properties in the selected metadata formats have been assembled in mapping tables[79]. For each metadata format, a mapping table with the following information has been created:

- The name of the property being mapped to.
- The semantic relation (exact match, more specific, more generic, or related).
- The name of the metadata format property.
- Details about how to do the mapping.
- The datatype of the metadata format property.
- When appropriate, an XPath 1.0 expression pointing to the property in the format.

Table 6.2 shows an excerpt of the mapping table for MPEG-7.

To implement the mappings described in the mapping tables there are two main approaches: (1) of expressing the mappings using a Semantic Web language and (2) using a pivot upper ontology. With respect to the first approach, there are two possibilities. One of them is to express mappings using SKOS, which provides constructs to formalize how concepts are related to each other. These constructs include skos:exactMatch, to express that two concepts are equivalent in most cases, skos:closeMatch, to express an equivalence valid in some cases, skos:narrowMatch and skos:broadMatch, to express hierarchical relationships between concepts, and skos:relatedMatch, to express any other type of relatedness. The second approach

[79] http://www.w3.org/TR/2012/REC-mediaont-10-20120209/#property-mapping-table, last checked December 18, 2013.

Table 6.2: Excerpt of the mapping table from Ontology for Media Resource 1.0 to MPEG-7

Property	Semantic relation	MPEG-7 XPath	Mapping
identifier	more specific	DescriptionMetadata/-PublicIdentifier or Media-Information/Media-Identification/EntityIdentifier	identifier:value; (Unique ID)
title	exact	CreationInformation/-Creation/Title	title:value; (string) type:@type; (anyURI)
language	exact	CreationInformation/-Classification/Language	language:value; (string)
locator	exact	MediaInformation/Media-Profile/MediaInstance/-MediaLocator/MediaUri	locator:value; (anyURI)

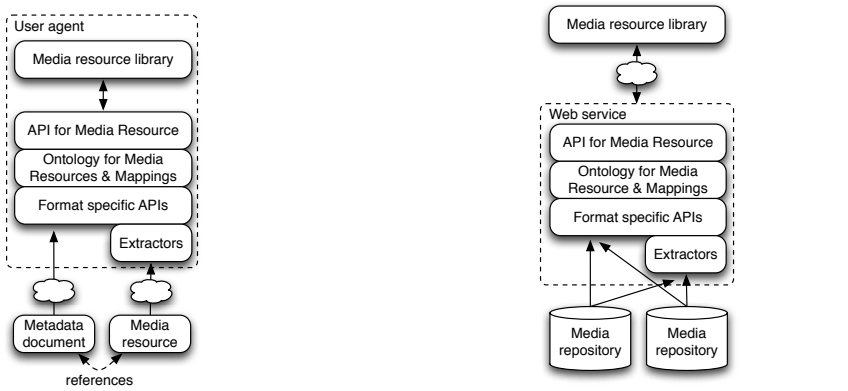
consists of expressing mappings using OWL and SWRL[80]. Regarding the second approach, there are also two different ways: (a) expressing mappings using a format independent ontology and (b) expressing mappings using built-in properties in an ontology directly related to the Media Ontology. Advantages and disadvantages of each approach are presented in [SBB+09].

6.3.3 API for Media Resources 1.0

The API for Media Resource 1.0 enables an interoperable access to metadata information related to media resources on the Web, with the defined core vocabulary as recommended best practice. Different design considerations have been discussed leading to the specification of global interfaces with specific parameter. This implicates a minimal number of exposed interfaces ensuring a broad adoption and less security leaks. Further, it reduces implementation work while designing applications or integrating the API into legacy systems.

The API can be used in two modes of operation: asynchronous and synchronous mode. For this API the asynchronous mode is considered to be used as default, where calls return without waiting for the request to finish its execution: a call-back function is provided to be invoked when the request terminates. On the other hand, synchronous calls wait for the request to terminate and directly return the result. The API is considered to be used in two scenarios as illustrated in Figure 6.2.

[80] http://www.w3.org/Submission/SWRL/, last checked December 18, 2013.



(a) Client only implementation (b) Client-server implementation

Figure 6.2: Deisgn considerations of the API for Media Resource

In the first scenario the API is encapsulated in a user agent acting as client only implementation whereas in the second it is implemented as a Web service, more specifically a client-server implementation.

The API consists of two main parts: (i) interfaces to access media resources and (ii) a set of core properties describing the information in an interoperable way along with their JSON serialization. The API is defined using WebIDL[81], which is an IDL variant explicitly covering programming languages commonly used on the web (e.g., ECMAScript).

Listing 6.1 shows the three central interfaces to get access to metadata information of media resources. `MediaResource` (line 1–8) is the basic constructor offering the possibility to get information of the mode of operations available. Following, there are two interfaces inherting from `MediaResource`, one for each mode of operations, namely `AsyncMediaResource` (line 10–24) and `SyncMediaResource` (line 26–36). In general both offer the `getOriginalMetadata` and `getMediaProperty` methods. The first method returns metadata of an associated media resource in the underlying metadata format. In contrast to that, `getMediaProperty` offers the possibility to define a subset of the core properties for which metadata information should be retrieved. It is further possible to specify filter criteria, e.g., to get only metadata annotations for a specific language.

Listing 6.2 illustrates the property definition of the API for Media Resource 1.0. Each property implements the generic interface `MediaAnnotation` (line 1–9 in Listing 6.2) to inherit basic attributes, such as `propertyName`, `value` or its `sourceFormat`. Following the ontology design, every property has the possibility to carry its metadata information in an unstructured (only attributes of `MediaAnnotation` in use) as well as a structured way (property specific attributes). As an example, the definition for *title* has been also added in Listing 6.2 (line 11–15). The structured values are divided into two attributes carrying the explicit value and

[81] `http://www.w3.org/TR/WebIDL/`, last checked December 18, 2013.

a URI referencing the semantic concept defined by a controlled vocabulary.

Besides the formal API specification, its behavior is also defined by the use of a specific subset of the HTML/1.1 status codes[82]. The complete WebIDL specification of this API can be found in Listing B.1 of Appendix B.2.

[82] http://www.w3.org/TR/2011/WD-mediaont-api-1.0-20111122/#api-status-codes, last checked December 18, 2013.

```
1  interface MediaResource {
2    short getSupportedModes();
3    MediaResource createMediaResource(
4        in DOMString                  mediaResource,
5        in optional MetadataSource[] metadataSources,
6        in optional short             mode
7    );
8  };
9
10 interface AsyncMediaResource : MediaResource {
11   void getMediaProperty(
12       in DOMString[]        propertyNames,
13       in PropertyCallback   successCallback,
14       in ErrorCallback      errorCallback,
15       in optional DOMString fragment,
16       in optional DOMString sourceFormat,
17       in optional DOMString language
18   );
19   void getOriginalMetadata (
20       in DOMString        sourceFormat,
21       in MetadataCallback successCallback,
22       in ErrorCallback    errorCallback
23   );
24 };
25
26 interface SyncMediaResource : MediaResource {
27   MediaAnnotation[] getMediaProperty(
28       in DOMString[]        propertyNames,
29       in optional DOMString fragment,
30       in optional DOMString sourceFormat,
31       in optional DOMString language
32   );
33   DOMString[] getOriginalMetadata (
34       in DOMString sourceFormat
35   );
36 };
```

Listing 6.1: Central interfaces of the API for Media Resource 1.0

```
1  interface MediaAnnotation {
2      attribute DOMString propertyName;
3      attribute DOMString value;
4      attribute DOMString language;
5      attribute DOMString sourceFormat;
6      attribute DOMString fragmentIdentifier;
7      attribute DOMString mappingType;
8      attribute short     statusCode;
9  };
10
11 interface Title : MediaAnnotation {
12     attribute DOMString titleLabel;
13     attribute DOMString typeLink;
14     attribute DOMString typeLabel;
15 };
```

Listing 6.2: Property definitions of the API for Media Resource 1.0

Chapter 7

Discussion

The prior chapter discussed the interoperability issues of multimedia metadata schemas on the Web[83]. As mentioned, the proposed specifications *Ontology* and *API for Media Resource 1.0* are designed in a domain independent way and are therefore utilized in both real-world application scenarios of Section 1.1. Before discussing the improvements of the specifications in terms of the superordinate real-world application scenarios, their appearance in scientific research projects will be given to underline the postulated domain independence. In this light, Table 7.1 gives an overview of current scientific projects implementing the two specifications. While the table clearly indicates that the projects exhibit diverse overall aims, all of them were able to integrate the specifications following different layers of conformance.

The *PrestoPRIME Semantic Converter* [HBNM11] is an automated metadata mapping service for audiovisual metadata in the archival domain that uses the Ontology for Media Resources as an interoperable target format. It supports a number of metadata standards and proprietary formats of archive or broadcast organizations can be added. The use case of making archive content accessible on the Web – together with its metadata – is becoming increasingly important, making the *Ontology for Media Resources* a relevant target format for publication and interoperability with Linked Data.

The *Multimedia Metadata Ontology (M3O)* [SES12] is a comprehensive model for representing multimedia metadata, based on the foundational ontology DOLCE+DnS Ultralight and several ontology design patterns, which has been aligned to the *Ontology for Media Resources*. M3O serves as generic modeling framework for integrating the existing metadata models and metadata standards rather than replacing them, providing also support for the *Ontology for Media Resources*.

The *Linked Media Framework* [KSB11] is an easy-to-setup server application that bundles central Semantic Web technologies to offer publishing legacy data as linked data, building semantic search applications and enabling information extraction. A number of additional modules are provided in the LMF, among them the Media Interlinking module, which uses the *Ontology for Media Resources* and the W3C URI for Media Fragments to enable integration with heterogeneous data sources on the Web.

NinSuna [LDMW11] is a metadata-driven media adaptation and delivery framework, making use of novel media support in HTML5 and also supporting fragment-

[83] This Chapter is partially based on [SBB+13].

Table 7.1: Projects utilizing the Ontology and API for Media Resource 1.0

Project name	*Type*	*Ontology*	*Mappings*	*API*
PrestoPRIME Semantic Converter [HBNM11]	metadata mapping service	✓	✓	✗
Multimedia Metadata Ontology [SES12]	metadata mapping service	✓	✗	✗
Linked Media Framework [KSB11]	portal service	✓	✓	✓
NinSuna [LDMW11]	portal service	✓	✓	✗
EventMedia [TMF10]	portal service	✓	✗	✗

based access conforming to the W3C URI for Media Fragments specification. Metadata for the media items is published in RDF conforming to the *Ontology for Media Resources*, providing powerful time- and region-based annotation capabilities in combination with fragment identifiers. The use of the Ontology for Media Resources is a driver for interoperability and allows the integration of other data sources within NinSuna.

EventMedia [TMF10] aggregates a large dataset composed of event descriptions (from the public event directories last.fm, eventful and upcoming) together with media descriptions associated with these events and interlinked with the larger Linked Open Data cloud. A Web-based environment allows users to explore and select events and to view associated media. The *Ontology for Media Resources* has been used for representing the metadata of these media and has enabled interlinking with the Linked Open Data cloud.

All of the presented projects use the core vocabulary in order to describe the information of a media resource in a unified way. In addition to the support of the core vocabulary, the PrestoPRIME Semantic Converter, the Linked Media Framework and NinSuna also implement the mappings defined by the group (e.g., realized by XSLT style sheets). These applications thus bridge the interoperability gap by providing multimedia metadata published in different source formats in an unified way. The Linked Media Framework furthermore implements the API to enable a unified retrieval over the heterogenous landscape of metadata formats available in the Linked Data cloud.

In addition to these projects, a starting point for future implementations focusing on the *API for Media Resources 1.0* are the following two open source showcases[84]: the first deals with an image gallery showing images as well as its metadata information. Here, the API is implemented as a Web service following the synchronous mode of operations. In contrast to that, the second showcase

[84]Both implementations are available online at: `http://mawg.joanneum.at/`

utilizes the API in a browser extension following the asynchronous mode of operation. The application enables a user to generate a video playlist, where videos and the corresponding metadata information from different platforms can be arranged in an unified way. These implementations serve as a validation for the API specification, which provided useful feedback for the specification and confirmed its implementability. In addition, the code of these implementations provides a convenient starting point for developers interested in implementing the API. Both implementations have been created under my supervision.

In the context of this thesis, the improvements enabled by the specifications tackle the metadata interoperability issues in the following way: in terms of the federated medical retrieval use case metadata interoperability only plays a minor role. Nevertheless, the pivot metadata schema is utilized to create a loose federated schema[85] of the underlying data. Here, the *identifier* property expresses the semantic links between the present retrieval services. Within the distributed query execution workflow, those links are foster federated retrieval abilities and are needed during query planning and aggregation of partial result sets. Besides, the isolated image retrieval use case heavily depends on the application of both specifications. In this domain, the pivot metadata scheme enables harmonization between the diverse metadata-based retrieval services each of them enforcing the usage of diverse (proprietary) metadata schemas. The usage of a unified description scheme, the mapping tables allow an on-the-fly syntactical mapping of metadata instances between the pivot metadata model and the present metadata schemas in each retrieval service. Finally, the API extends the retrieval services to ensure an homogenous access to the encompassed media resources by client applications.

[85] Details on the query execution and the loose federated schema are given in Section 8.7.

Part IV

Distributed Multimedia Retrieval

Chapter 8

AIR: Architecture for Interoperable Retrieval

This chapter[86] proposes an architecture to enable *unified multimedia retrieval*, the second contribution of this thesis. In this light, the basic concepts of the *architecture for interoperable retrieval on distributed and heterogeneous multimedia repositories (AIR)* [SDK+10] are introduced. AIR is a full fledged multimedia retrieval system, especially designed by me to operate in heterogeneous multimedia retrieval environments. Related work to this topic is presented in Section 8.1. The generic real-world application scenarios of Section 1.1 are presented with concrete configurations Section 8.2. Section 8.3 gives insights to the underlying design principles whereas an excursus into utilized concepts of MPQF can be found in Section 8.4. AIR is equipped with two different query execution strategies, which are presented in Section 8.5. Its architectural facets are part of Section 8.6 whereas a consideration of distributed query processing in Section 8.7 conclude the chapter.

8.1 Related Work

Several approaches for accessing multimedia data in a possibly distributed and heterogeneous environments have been proposed:

In 2002, Löffler et al. [LBEK02] proposed a multimedia retrieval and indexing framework called *IFINDER*. It was able to process both, video and audio data, to generate multimedia metadata information on the basis of MPEG-7. Though the system was built to interconnect several backends, it has not been designed to deal with interoperability problems on the modeling level or a unified retrieval. Möller et al. issued in [MS07] a generic framework for medical search and retrieval. The application consists of a graphical metadata extractor, an annotation interface and a search interface. Here, the search interface is rather limited regarding the multimedia search capabilities and the metadata extractor is closed to the DICOM standard, but it is able to address heterogeneous data sources. More recently, Tous et al. [TD09] proposed an architecture for search and retrieval of still images. This architecture is based on three main components, covering the query format, the file transfer and registration of metadata ontologies. At its core, these interfaces use international standards, such as the MPEG Query Format. Unfortunately, this system is not able to deal with heterogeneous data sources.

[86] This Chapter is partially based on [SDK+10] and [SSB+12].

There are also approaches directly focusing on the heterogeneous nature of retrieval environments. Garcia et al. [GC05] build a multimedia retrieval framework on the basis of Semantic Web technologies. They used a MPEG-7 ontology as well as an SQL dialect to enable interoperability. As already outlined in this thesis, the conversion of an XML-based metadata schema to RDF does not solve the interoperability issue between different schemas nor does it extend semantics. Chen et al. [CCL08] issued a retrieval system that is utilizing social trust analysis to aggregate result sets of various autonomous retrieval services. Due to the lack of a standardized query language as well as multimedia metadata schemas, the possibility to create complex, flexible and multimedia-aware querying are very limited. Laborie et al. [LMS10] follows a similar way with the LINDO project as the AIR framework. Here, they are focusing on the creation of a single generic metadata schema to soften the interoperability issues on the modeling level.

AIR, as it is described in this thesis, utilizes the findings described in [DBKG08]. However, it enriches these concepts in several means: component-based architecture, metadata interoperability, various query execution strategies as well as optimization of query execution. The consideration of the concepts described in Section 8.3 also had a deep impact regarding the architecture of AIR. This makes it possible to tailor AIR specifically to the needs of a specific use case.

8.2 Application Scenarios

The current prototype of the AIR framework has been integrated in two specific configurations of the real-world application scenarios introduced in Section 1.1. Recapitulatory they have been selected, because they differ i) in their covered domain and ii) in the way, the query is being processed. This diversity clearly shows the applicability of the framework in a wide range of usage scenarios. Here, *THESEUS: MEDICO* is the implementation of the federated medical retrieval system whereas the *Interoperable Image Search* belongs to the isolated image retrieval system use case.

8.2.1 THESEUS: MEDICO

The *THESEUS* project[87] is funded by the German Federal Ministry of Economics and Technology. Its challenge is to find ways of providing users with simple and efficient access to this enormous amount of knowledge available on the Web. The applications of this project should develop new mechanisms for automatic annotation of data, rapid processing of multimedia documents or innovative ontology management. The main project is subdivided in six sub-projects, that are settled in a variety of domains (e.g., digital libraries). The mission of the *MEDICO* application scenario is to establish an intelligent and scalable search engine for the medical domain by combining medical image processing and semantically rich image

[87] http://theseus.pt-dlr.de/, last checked December 18, 2013.

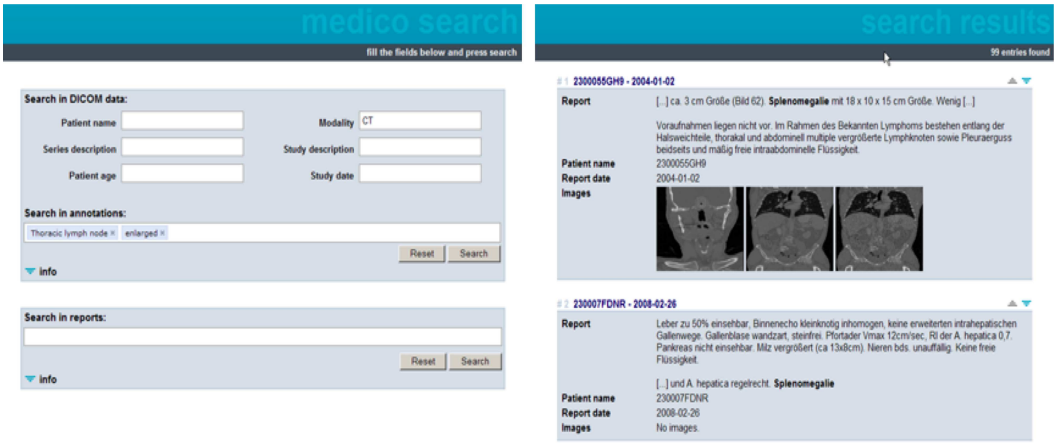


Figure 8.1: *THESEUS: MEDICO* web interface for retrieval tasks

annotation vocabularies.

Figure 8.1 sketches an end-to-end workflow inside the *MEDICO* system. It provides the user with an easy-to-use web-based form to describe his/her search query. Currently, this user interface utilizes a semantically rich data set composed of DICOM tags, image annotations, text annotations and gray-value based 3D CT images. This leads to a heterogeneous multimedia retrieval environment with multiple query languages: DICOM tags are stored in a PACS system, image / text annotations are saved in a triple store and the CT scans are accessible by an image search engine performing a similarity search. Apparently, all these retrieval services are using their own query languages for retrieval (e.g., SPARQL) as well as the actual data representation for annotation storage (e.g., RDF/OWL). Beside all differences, these different data sources describe a common (semantically linked) global data set. To fulfill a meaningful semantic search, the present interoperability issues have to be solved. Furthermore, it is essential to formulate queries that take the aforementioned diverse retrieval paradigms into account. For this purpose, *MEDICO* integrates the AIR multimedia middleware framework, following the federated query processing strategy as described in Section 8.5.

8.2.2 Interoperable Image Search

In contrast to *THESEUS: MEDICO*, Figure 8.2 shows the image retrieval system, which consists of three independent parts. The retrieval process is based on the already mentioned JPSearch standard, issued by ISO/IEC SC29 WG1 (commonly known as JPEG). Within this standard, a specific query language – JPEG Query Format (JPQF) – has been defined, which is using a subset (tailored to image retrieval) of the MPEG Query Format (MPQF) [DTG+08]. In the following, the different parts will be highlighted from a functional point of view.

The data source is a heterogeneous image retrieval environment, whereas retrieval services act autonomous. In this context, autonomous means that the en-

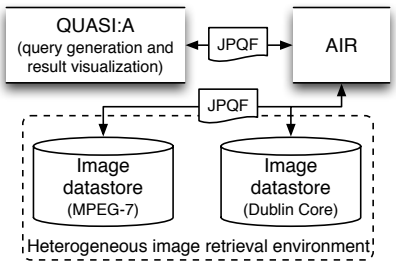
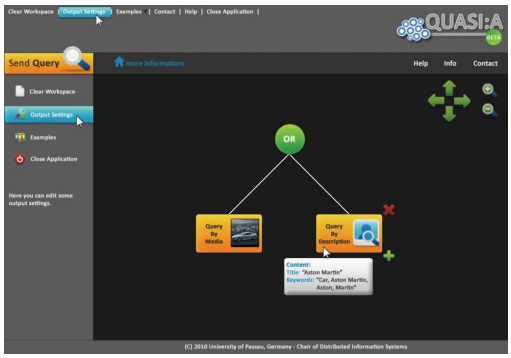


Figure 8.2: Overview of Interoperable Image Search environment



(i) Visual query generation

(ii) QueryByMedia functionalities

Figure 8.3: JavaFX based user interface QUASI:A

gaged retrieval servces have no direct correlation/connection in the first place. The following assumptions are made: the data stores feature retrieval services in order to process the incoming JPQF query as a whole (no segmentation of queries needed). Furthermore the image data sets may overlapping, but are annotated with diverse metadata formats, here MPEG-7 and Dublin Core. Therefore, duplicate elimination plays only a minor role in the aggregation process. The main challenge is to manage heterogeneity that is expressed by (i) different metadata formats for annotation and (ii) different query languages for retrieval.

The *query and search for images application (QUASI:A)* is JavaFX based and supposed to offer JPQF query generation, cf. Figure 8.3 (i), as well as result presentation functionalities. As a proof of concept, only a subset of JPSearch functionalities has been implemented, focusing on the specified interoperability issues. Therefore, it is restricted to the three JPQF query types: QueryByMedia, QueryByDescription and QueryByRelevanceFeedback. The first query type is an implementation of the well-known query by example paradigm. Here, a user is able to specify a picture (e.g., accessible on the internet or via file upload) that serves as an input for a similarity search. This picture can also be modified (e.g., crop or resize), as shown in Figure 8.3 (ii), where a special region of interest has been selected. The second query offers the possibility to define a metadata based search. Here, a user may fill out a form containing elements of the already introduced pivot metadata

schema to perform an exact metadata search. These query types and the comparison types can be linked by the use of Boolean operators (e.g., AND) in a tree based manner, as illustrated in Figure 8.3 (ii). This visualization technique ensures clarity and usability. The images stored in the aggregated result set will be presented in a gallery fashioned way. Here, a single image of the gallery can be directly used as an input for a further similarity search (browsing) or a subset (positive as well as negative examples) of the result set defining a relevance feedback query. In this use case, the AIR framework interconnects those parts and serves as mediator.

8.3 Design Principles

Besides international standards, interoperable media retrieval can be established by the introduction of a middleware system abstracting the communication, namely a *broker*. A broker directly corresponds to the proposed interoperable systems architecture of Section 5.2. In this sense, it acts as mediator between multimedia clients and retrieval systems improving its collaboration remarkable. It accepts complex multi-part and multimodal queries from one or more clients and maps/distributes those to multiple connected multimedia retrieval systems. As a fact, implementation complexity is reduced at the client side as only one communication partner needs to be addressed. However, the actual retrieval process of the multimedia data is performed inside the connected data stores. To ensure interoperability between the query applications and the registered retrieval services, the mediator is based on the following design principles:

- *Query language abstraction:*
  AIR is capable to federate an arbitrary amount of retrieval services utilizing various query languages/APIs (e.g., XQuery, SQL or SPARQL). This is achieved by converting all incoming queries into an internal abstract format that is finally translated into the respective specific query languages/APIs of a data store. As an internal abstraction layer, AIR makes use of the MPQF, which supports most of the functions in traditional query languages as well as several types of multimedia specific queries (e.g., temporal, spatial, or query-by-example). Further details on the query language can be found in Section 8.4.

- *Multiple retrieval paradigms:*
  Retrieval systems are not always following the same data retrieval paradigms. Here, a broad variety exists, e.g. relational, NoSQL[88] or XML-based storage or triple stores. AIR attempts to shield the applications / users from this variety. Further, it is most likely in such systems, that more than one data store has to be accessed for query evaluation. In this case, the query has to be segmented and distributed to applicable retrieval services. Following this, AIR acts as a federated database management system.

[88] http://nosql-database.org/, last checked December 18, 2013.

- *Metadata format interoperability:*
  As already outlined, a plethora of metadata formats are applied to describe syntactic or semantic attributes of media resources. Currently, there exist a huge number of standardized and proprietary metadata formats for nearly any application domain. Thus, more than one metadata formats are in use in a heterogeneous retrieval scenario. AIR therefore provides functionalities to perform the transformation between diverse metadata formats where a defined mapping exists and is made available.

- *Modular architectural design:*
  A modular architectural design should be always a fundament in software development. The central aspects in these topics are convertibility, extensibility and reusability. These ensure that the components are loose coupled in the overall system supporting an easy extension of the provided functionality of components or even the replacement of these by new implementations.

This set of design principles has been carefully reflected in the architecture of AIR. Before introducing the composition of AIR from a component based point of view, an excursus in MPQF will be given.

8.4 Excursus: MPEG Query Format

MPQF became an international standard in early 2009 as part 12 of the MPEG-7 standard. The main intention of MPQF is to formulate queries to address and retrieve multimedia data, like audio, images, video, text or a combination of these. At its core, MPQF is a XML-based query language specified by a XML Schema definition and intended to be used in a distributed multimedia retrieval environments. In addition, MPQF adds support for asynchronous search requests as well. In contrast to a synchronous request (immediate answer), a user is able to define a time period in an asynchronous scenario in which the result can be retrieved by any client. Beside the standardization of the query language, MPQF specifies a protocol for service discovery and service capability description. Here, a service is a particular system offering search and retrieval abilities.

Figure 8.4 shows the basic structure of the MPQF schema definition. The root element of a MPQF instance is always the `MPEGQuery` element. Depended on the actual client request, the query inherits a `Query` or a `Management` element. The `Query` object specifies the actual query (Input Query Format, `Input` element), a possibility to fetch results of an asynchronous query (`FetchResult` element) or holds the result of a query (Output Query Format, `Output` element). In contrast to that, the `Management` element copes with the task of searching and choosing desired multimedia services for retrieval and is also split in an `Input` and `Output` element.

The actual structure of the Input Query Format (IQF) is shown in Figure 8.5. It provides means for describing query requests from a client and carries the (aggregated) results from the service(s). In detail, the Input Query Format can

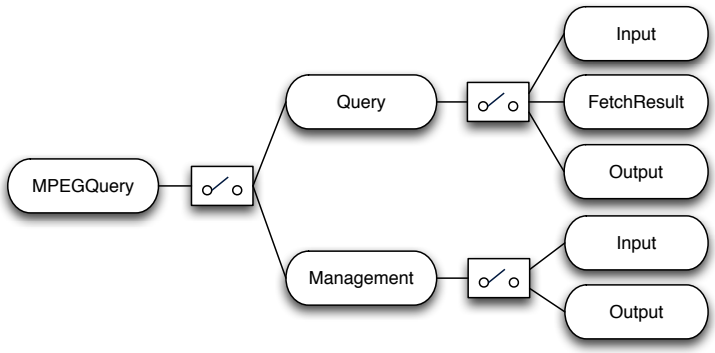


Figure 8.4: Core elements of MPQF

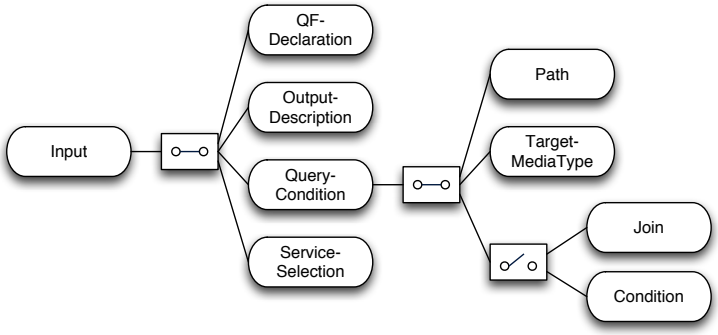


Figure 8.5: Structure of the Input Query Format

be composed of three different parts. The first is a optional declaration part, `QFDeclaration` element, pointing to resources (e.g., image file or its metadata description, etc.) that are used within the `QueryCondition` element or `Output-Description` part. The `OutputDescription` part allows the definition of the structure as well as the content of the expected result set. In distributed retrieval environments, the `ServiceSelection` enables an automatic routing of a user request to designated services. Finally, the `QueryCondition` element denotes the search criteria. It arranges a set of different query types (see Table 8.1) and expressions (e.g., `GreaterThan`), which can be combined by Boolean operators (e.g., `AND`). In general, all those query operands inherit from the `BooleanExpression` element as

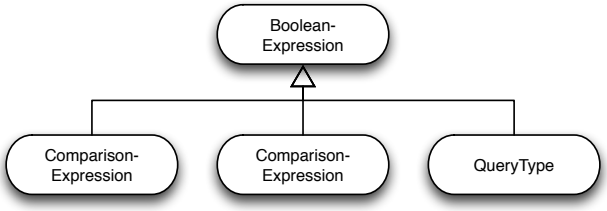


Figure 8.6: Relationship between query operators

illustrated in Figure 8.6. Further, the type of media resources can be restricted to a specific MIME type.

Table 8.1: Overview of MPQF query types

Query type	**Description**
QueryByMedia	Similarity or exact search using query by example
QueryByDescription	Similarity or exact search using XML-based metadata
QueryByFreeText	Free text retrieval
QueryByFeatureRange	Range retrieval, e.g., for low level features
SpatialQuery	Retrieval of spatial elements within media objects (e.g., person in a still image)
TemporalQuery	Retrieval of temporal elements within media objects (e.g., a scene in a video)
QueryByXQuery	Container for limited XQuery expressions
QueryByRelevanceFeedback	Retrieval that takes ranked result items of a previous search into account
QueryByROI	Retrieval based on a certain region of interest
QueryBySPARQL	Container for SPARQL queries

The structure of the Output Query Format (`Output` element) is introduced in Figure 8.7. The `GlobalComment` and `SystemMessage` element in general carry status information of services about the conducted retrieval. The actual results are stored in the `ResultItem` element. This is split up in several subelements, further specifying the retrieved data. This set contains, e.g., `TextResult` element, the `MediaResource` element (carrying the actual Base64 data of the resource or a URI to look it up) or a `Description` element.

As already mentioned, the management part of MPQF is used to handle information over a set of services and their retrieval abilities. Figure 8.8 depicts the element hierarchy of the management tools in MPQF. This includes service discovery, querying for service capabilities and service capability descriptions. The management part of the query format consists of either the Input or Output element depending on the direction of the communication (request or response). In

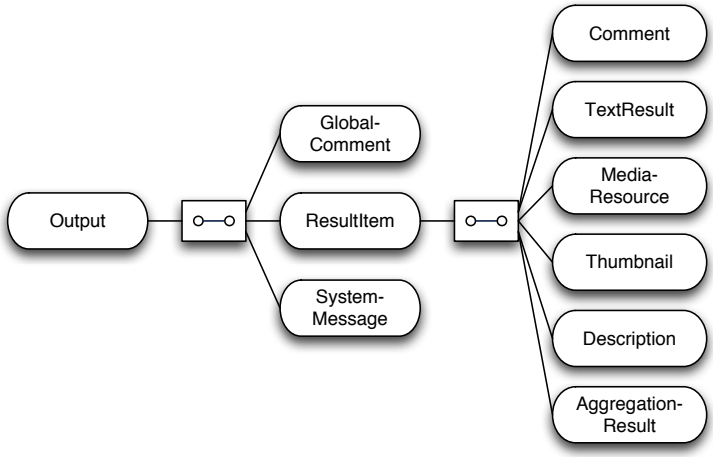


Figure 8.7: Structure of the Output Query Format

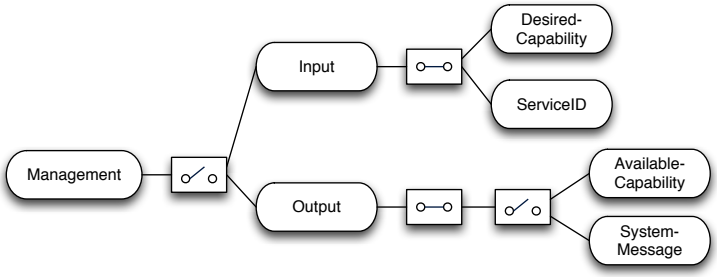


Figure 8.8: Structure of the management part of MPQF

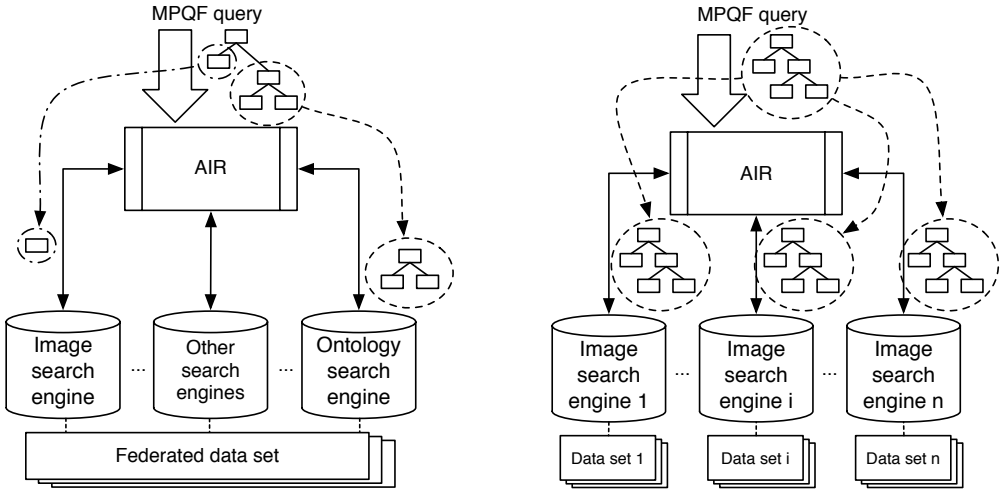
detail, a service may register its retrieval capabilities at the server / mediator with the `DesiredCapability` and its entry point in `ServiceID`.

This consideration only introduced a subset of the overall MPQF features available in the standard. This focus directly corelates with the implemented MPQF features in the AIR framework.

8.5 Query Execution Strategies

The AIR framework can be operated in many different facets within a distributed and heterogeneous multimedia search and retrieval framework. In general, the tasks of every internal component highly depend on the registered databases and use cases. In this context, two main query processing strategies have to be distinguished, as illustrated in Figure 8.9. Both of them cover the needs identified for distributed multimedia architectures presented in Section 5.2.

Federated Query Processing. The first paradigm deals with registered and participating retrieval systems that allow distributed processing on the basis of a federated data set, see Figure 8.9 (i). The involved heterogeneous systems may



i) Federated query processing ii) Autonomous query processing

Figure 8.9: AIR query processing strategies

depend on different data representation (e.g., ontology based semantic annotations and XML based low level features) and query interfaces (e.g., SPARQL and XQuery) but describe a common (linked) global data set. In this context, a query transmitted to AIR needs to be evaluated and optimized with regards to the overall query execution plan. In series, the initial query will be segmented and those are forwarded to the respective engines and are there executed. Now, the result aggregation has to deal with a correct consolidation of the partial result sets. In this context, AIR acts as a federated multimedia database management system.

Autonomous Query Processing. The second paradigm deals with registered and participating retrieval systems that are able to process the whole query locally, see Figure 8.9 (ii). In this sense, those heterogeneous systems provide their local metadata format (e.g., Dublin Core, MPEG-7, etc.) and a local / autonomous data set. A query transmitted to such systems is understood as a whole and the items of the result set are the outcome of an execution of the whole request. Of course, transformation of the used metadata format (e.g., from Dublin Core to MPEG-7) may be needed for some systems. In addition, depending on the degree of overlap among the data sets (e.g., the same image is annotated in all services), the individual result sets may contain duplicates. However, a result aggregation process needs to perform an overall ranking of the result items of the involved retrieval systems. Here, duplication elimination algorithms are applied as well.

Looking into both processing strategies, AIR is acting as a mediator system managing the entire retrieval environment and query workflow. Obviously, the single retrieval services are treated as black boxes. Following this observation, AIR strictly falls into the category of an external metasearch engine. Detailed examples

for both query processing strategies can be found in Section 8.7.

8.6 Architectural Facets

Figure 8.10 illustrates an end-to-end workflow scenario in a distributed retrieval scenario. AIR transforms incoming user queries (of different formats) to a common internal representation (MPQF) for further processing and distribution to registered data resources. It aggregates the returned results before delivering it to the client. AIR is able to handle synchronous as well as asynchronous queries. In the following, the subcomponents of AIR are briefly described, which are illustrated in Figure 8.10. From a technical point of view, AIR is entirely written in Java[89] and integrated into the Spring framework[90] to configure the entire processing chain and the application life cycle as well as Apache Maven[91] for quality control.

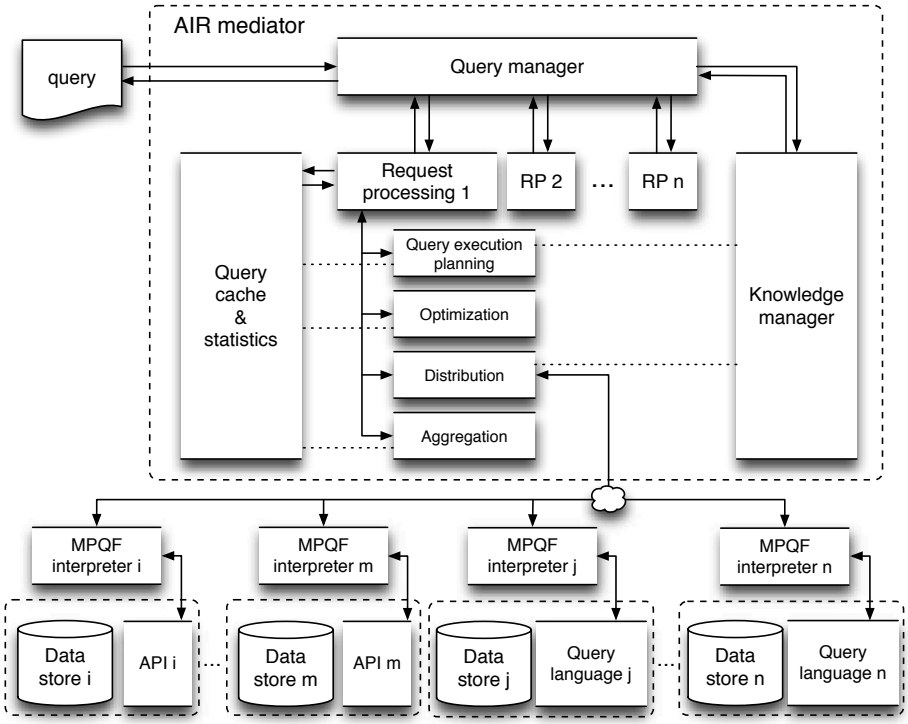


Figure 8.10: Architectural overview of the AIR mediator framework

The *QueryManager* is the entry point of every user request. Its main purpose is the receiving of an incoming query as well as API assisted MPQF query generation and validation of MPQF queries. In case an application is not aware in formulating MPQF queries, these can be build by consecutive API calls. Following this, two

[89] http://www.java.com/, last checked December 18, 2013.
[90] http://www.springsource.org/, last checked December 18, 2013.
[91] http://maven.apache.org/, last checked December 18, 2013.

main parts of the MPQF structure will be created: First, the QueryCondition element holds the filter criteria in an arbitrary complex condition tree. Second, the OutputDescription element defines the structure of the result set. In this object, the needed information about required result items, grouping or sorting is stored. After finalizing the query creation step, the generated MPQF query will be registered at AIR using the query cache and statistics component In case an instance of a query is created at the client side in MPQF format then this query will be directly registered at AIR. After a query has been validated, it is forwarded to its destination, in names the KnowledgeManager or the RequestProcessing component.

The main functionalities of the *KnowledgeManager* are the (de-)registration of data stores with their capability descriptions and the service discovery as an input for the distribution of (sub-)queries. These capability descriptions are standardized in MPQF, allowing the specification of the retrieval characteristics of registered data stores considering for instance the supported query types or the underlying metadata formats. In series, depending on those capabilities, the KnowledgeManager is able to filter registered data stores during the search process (service discovery). For a registered retrieval system, it is very likely that not all functions specified in the incoming queries are supported. In such an environment, one of the important tasks for a client is to identify the data stores, which provide the desired query functions or support the desired result representation formats identified by e.g., an MIME type using the service discovery.

For each query a single *RequestProcessing* component will be initialized. This ensures parallelism as well as it guarantees that a single object manages the complete life cycle of a query. The main tasks of this component are query execution planning, creation and optimization of the chosen query execution plan, distribution of query and result aggregation. Besides managing the different states of a query, this component sends a copy of the optimized query to the query cache and statistics component, which collects information in order to improve optimization. Regarding the lifetime of a query, the following states have been defined: pending (query registered, process not started), retrieval (search started, some results missing), processing (all results available, aggregation in progress), finished (result can be fetched) and closed (result fetched or query lifetime expired). These states are also valid for the individual query segments, since they are also valid MPQF queries.

The *Query cache and statistics* component organizes the registration of queries in the query cache. It collects information about data stores, such as execution times, network statistics, etc. Besides the data store statistics, the complete query will be stored as well as the partial result sets. The information provided by this component will be used for two different optimization tasks, namely: internal query and query stream optimization. Internal query optimization is a technique following well-known optimization rules of the relational algebra (e.g., operator reordering on the basis of heuristics / statistics). In contrast to that, query stream optimization is intended to detect similar / equal query segments that have already been evaluated. If such a segment has been detected, the results can be directly injected into the

query execution plan. Obviously, the query cache will also implement the paging functionality. Section 10 will introduce this optimization techniques in more detail.

Finally, *MPQF interpreters* act as a mediator between AIR and a particular retrieval service. An interpreter receives an MPQF formatted query and transforms it into native calls of the underlying query language of the database or retrieval engine. The actual retrieval is done by the specific services. In this context, several interpreters (mappers) for heterogeneous data stores have been implemented (e.g., Flickr, XQuery, etc.). After a successful retrieval, the Interpreter converts the result set in a valid MPQF formatted response and forwards it to AIR.

8.7 Distributed Query Processing

In general, the query execution plan in a distributed multimedia retrieval systems can be aligned to the phases defined for distributed database systems [Dad96].

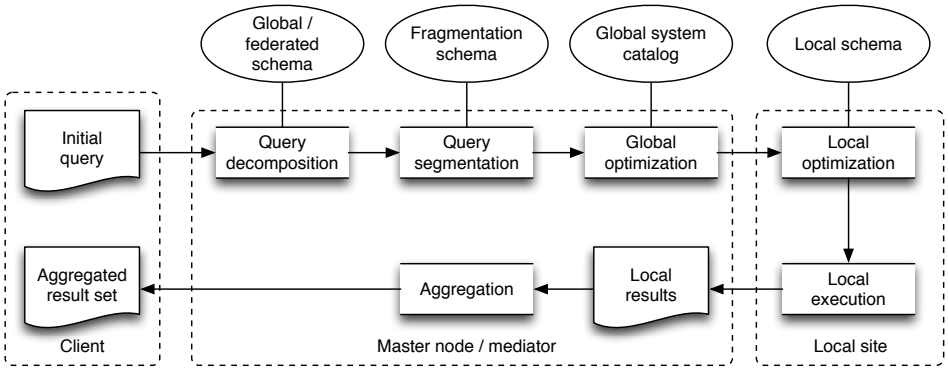


Figure 8.11: Phases of distributed query processing

Figure 8.11 illustrates the different phases along with their execution environment. Typically, in a distributed retrieval environment there exists at least one client, a master node / mediator as well as connected local sites, here retrieval services or local databases.

Query decomposition is the entry point for each initial query of a client request. The incoming query is analyzed and transformed in an internal presentation using the global schema. Crucial steps are syntactic and semantic parsing, algebraic simplification and normalization (conjunctive vs. disjunctive normal form).

Query segmentation performs a fragmentation of the transformed query into valid subqueries. By the help of the fragmentation schema the created subqueries can be distributed to the specific endpoints.

Global optimization specifies the execution strategy of query segments and is handled by the query optimizer. The optimizer consists of three different

parts: *search space*, *cost model* and the *search strategy*. The search space is defined by the set of possible and therefore equivalent query execution strategies. All strategies are defining the same query semantic and therefore equal result sets. The cost model calculates costs for a particular operation and accordingly for the complete query execution plan. In general, the formula defined by Lohmann et al. in [LMH+85] can be used as a basis to estimate these costs. Parameters of a cost model are typically I/O, CPU, network communication or local processing costs. Finally, the search strategy is the technique to select a specific plan using the cost model. Since a very large set of plans can be generated of a single query, an intelligent selection process ensures efficiency.

Local optimization and **local execution** means that each local site can restructure the subquery as well as choose specific (physical) implementations of operators on its own. This internal information is mostly not distributed to external peers.

Aggregation is again performed by the master node / mediator. Here, the single results are assembled to a single result set, which is valid to the overall query semantic. Depending on the overall characteristic of the retrieval environment (Heterogeneous vs. interoperable), join algorithms [Kos00] as well as fusion techniques [SF94] are applied to reach this goal.

Obviously, during those phases different processing algorithms are in use to manage multimedia data. The architecture for distributed multimedia retrieval presented in this thesis (c.p., Section 8) closely follows these processing steps. Further, it relies on the findings of Section 5.2 and is build as an interoperable system aware to handle federated environments as well.

To get an idea of general query processing workflows inside the AIR framework, for each processing strategy one example will show the complete processing chain.

Example (federated medical retrieval). The query specified in Section 1.1 for federated medical retrieval holds for the THESEUS: MEDICO use case and will serve as example for a federated multimedia query processing. Before looking into the query processing itself, the retrieval services are registered at the *KnowledgeManager* component of AIR. The register information consist of the entry point for the query, the retrieval abilities and a semantic link to other data stores. Obviously, the information of the semantic links is given by an domain expert leading to a federated retrieval environment as shown in Figure 8.12. Within this figure, three different IDs (*FindingUID, PatientID, SeriesInstanceUID*) are in use to determine equal semantic entities within the retrieval services. In this scenario, the *FindingUID* can be mapped to *SeriesInstnceUID*. Unification on the modeling level is reached by the utilization of the *identifier* property of the pivot metadata scheme. The retrieval services registered the following retrieval capabilities: the metadata-

based retrieval service is a Semantic Web triple store on the basis of Jena TDB[92] offering a SPARQL query interface. Therefore the registered MPQF query type is `QueryBySPARQL`. The content-based retrieval service abstracts from an underlying MySQL[93] database but the exposed API takes an media resource as input for the similarity search leading to the registration of a `QueryByExample` query type. Finally, DCM4CHEE[94] is an open source implementation of a PACS offering metadata-based retrieval with the DICOM standard as accepted input metadata scheme. The registration information consists of `QueryByDescription` and the namespace of the supported metadata scheme accordingly.

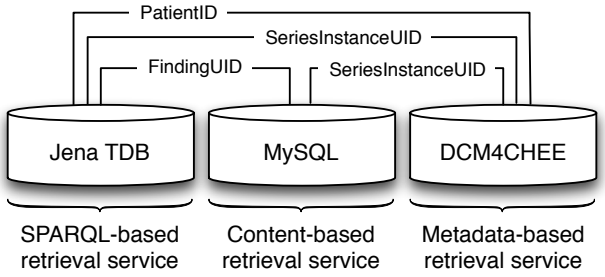


Figure 8.12: Federated retrieval environment of THESEUS: MEDICO

Within this environment, the following prose query will be executed:

> *"Give me all lesions, which are similar to a given region in an example CT scan, located inside the liver and the patient is at least of age 65!"*

It is clear, that the present information content of the prose query addresses all connected retrieval services. The web-based search client shown in Figure 8.1 creates a MPQF query by calling the API of the AIR framework, constituting the query decomposition phase. The resulting query is shown in Figure 8.13. The query structure is illustrated in Figure 8.13 and will be explained next: the `QueryByMedia` includes an URI to the example CT scan, the `QueryBySPARQL` addresses the ontology-based information *"located inside the liver"* whereas the `QueryByDescription` stores the information about "patient is at least of age 65". The three query types are connected by an Boolean `AND` operator and the projection is selecting the *FindingUID* expressing an actual lesion. The query processing chain continues as follows: within the query segmentation phase, the query is analyzed by focusing on its query leaves, which actually are the query types. Here, the AIR framework recognizes three different query types and calls the *KnowledgeManager* for applicable connected retrieval services. The *KnowledgeManager* is also in charge of validating and extending the semantic links to ensure a federated retrieval. In this case, *FindingUID* is a globally valid identifier and no enrichment

[92] `http://jena.apache.org/documentation/tdb/`, last checked December 18, 2013.
[93] `http://www.mysql.com/`, last checked December 18, 2013.
[94] `http://www.dcm4che.org/`, last checked December 18, 2013.

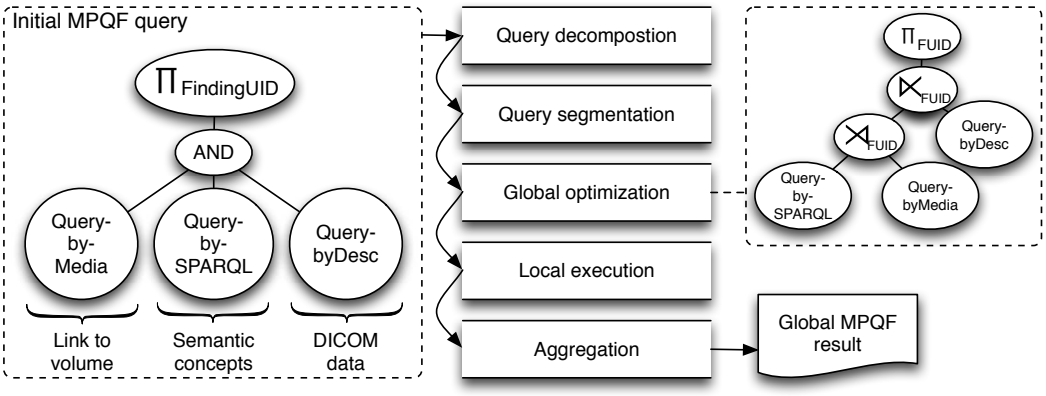


Figure 8.13: Federated query processing in the THESEUS: MEDICO use case

or changes have to be applied. Following the retrieved information, the query will be segmented in three partial queries each containing only one query type. The segments are arranged in an overall query execution plan ensuring the initial query semantics by the global optimization phase. Since the selection of the query is restricted to the *FindingUID*, *left* and *right outer joins* are used to minimize overall costs of the query execution. Between the global optimization and the local execution, the query segments are distributed to the MPQF interpreters located at the retrieval services and there executed internally. The partial result sets are returned to the AIR framework and the consolidation follows the query execution plan in the aggregation phase. The outcome is a single MPQF result, which will be returned in the appropriate format to the client application.

Example (isolated image retrieval). In contrast to the aforementioned example of federated retrieval, this example will shed light on the enrichment of a query during the query execution process. Let us consider a retrieval environment of five unconnected retrieval services (*retrieval service 1-5*). All of them are registered at the AIR framework with their retrieval capabilities. Here, *retrieval service 1* and *2* are associated with `QueryByMedia` and *retrieval service 3* and *4* with `QueryByDescription` utilizing the *Ontology for Media Resource 1.0*. *Retrieval service 5* has registered with a `QueryBySPARQL` query type. The environment is shown in Figure 8.14.

In our example, we consider the following prose query:

> *"Give me the first ten images that are similar to http://any.uri/strawberry.jpg or are annotated with the keyword strawberry!"*

As mentioned in Section 8.6, a query can be directly submitted to AIR as serialized MPQF query. The first step is query decomposition. In this phase, the query is analyzed and parsed into the internal object structure. Here, the MPQF query consists of two basic query types, namely `QueryByMedia` and `Query-`

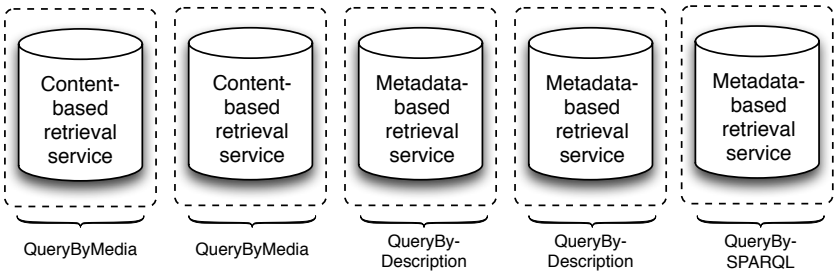
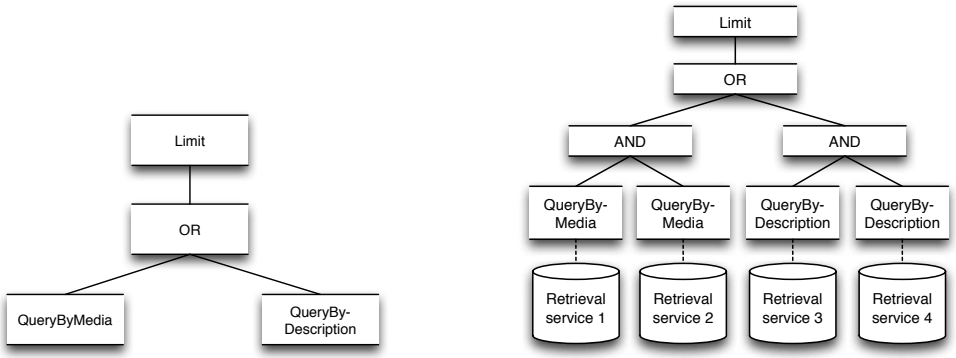


Figure 8.14: Retrieval environment of Interoperable Image Retrieval

`ByDescription`, that are concatenated with a Boolean `OR` operator. The `Limit` operator restricts the size of the selection to the given parameter, here ten.



i) Initial query ii) Extended and optimized query

Figure 8.15: Decomposition, segmentation and global optimization of an initial query

Figure 8.15 (i) shows the incoming query in a graphical representation of the query tree (not showing specific attributes of nodes). During the query segmentation phase, all leafs (query types) of the query tree are analyzed. This process is termed *service discovery* and a set of valid retrieval services for each single query type is generated. Beside an enriched version of the initial query, the outcome of this process are subqueries, that will be transmitted to the identified retrieval services. In this example, *retrieval services 1 to 4* are considered to be suitable to process certain subqueries whereas *retrieval service 5* will be ignored. The initial query will be extended by this knowledge without changing the overall retrieval semantics. The global optimization of the initial query determines the actual execution sequence of the subqueries generated by the query execution planning component. The remaining execution process of the query is equivalent to the example given beforehand.

Chapter 9

Discussion

This chapter introduced the AIR framework which targets on implementing interoperable multimedia retrieval in an profoundly heterogeneous environment by the use of standardized technologies. In this context, the framework used the newly developed MPEG Query Format for unifying multimedia retrieval requests. Besides, metadata heterogeneity is antagonized by the established by the specification and implementation of the Ontology / API for Media Resource 1.0 as well as JPSearch/JPEG. These features are completed by means for query management and distribution as well as service discovery and result set aggregation techniques.

Table 9.1: Comparison of frameworks enabling unified and interoperable retrieval

	Modularity	**Multimedia retrieval**	**Unified retrieval**	**Metadata interoperability**
IFINDER [LBEK02]	✗	✓	✗	✗
Möller et al. [MS07]	✓	✓	✓	✗
LEGO-like architecture [TD09]	✗	✓	✗	✓
Garcia et al. [GC05]	✓	✗	✗	✓
Chen et al. [CCL08]	✗	✗	✗	✓
LINDO [LMS10]	✗	✗	✗	✓
AIR	✓	✓	✓	✓

Since a comprehensive performance evaluation of the AIR framework is part of Chapter 12, the remaining discussion will focus on the unified retrieval and metadata interoperability aspect. The comparison in Table 9.1 is an summarization of Section 8.1 and clearly shows the advances of the proposed AIR architecture.

While the IFINDER and the system proposed by Möller et al. provide an unified access to heterogeneous data sources, it lacks in the expressiveness of proprietary multimedia queries and metadata interoperability. In contrast to that, the LEGO-like architecture sets the focus on metadata interoperability, but the heterogeneity problem remains unstudied. Further frameworks, such as proposed by Garcia et al., Chen et. al or LINDO focus on a specific sub-issue, here metadata interoperability. None of the named frameworks take all dimensions into account needed to establish a unified and interoperable retrieval in heterogeneous and distributed multimedia environments.

An evaluation regarding the retrieval abilities of the MEDICO system can be found in [STS+11]. Here, AIR is not in charge of performing the actual retrieval, such as similarity computations of the CT scans, but acts as a query federation service. It is able to transform and validate the incoming query into an internal MPQF abstraction and into the particular query languages or API calls. Further, it routes the query segments to the specific endpoints. Finally, it aggregates the partial results into a single result, fully compliant to the initial query semantic as shown in Section *sec:mmsysQryProc*. A more detailed consideration of the federation process tailored to the *THESEUS: MEDICO* application scenario can be found in [SDS+11].

Part V

Optimizing Distributed Multimedia Retrieval

Chapter 10

Optimization Techniques for Query Execution

The AIR framework as presented in the prior chapter aims towards unified multimedia retrieval. As shown in Section 8.7, there exist specific phases of distributed (multimedia) retrieval one has to address to guarantee a semantically correct as well as efficient query execution.

This chapter[95] focuses on this task by introducing the first part of the third contribution of this thesis, namely the optimization techniques for multimedia retrieval. This techniques have been implemented and evaluated[96] in the context of the AIR framework and its usage scenarios. The chapter is structured as follows: Section 10.1 gives insight into related work whereas Section 10.2 focuses on inter-query optimization. After the consideration of *query execution planning* in Section 10.2.1, two different *query processing strategies* (see Section 10.2.2) are highlighted. *Inter-query optimization* are discussed in Section 10.3. In this terms, intra-query optimization techniques focus on the improvement of the execution of a single query. In contrast to that, inter-query optimizations take parallel query streams into account.

10.1 Related Work

The multimedia database community has focused on the improvement of (distributed) multimedia query processing by adopting and designing optimization strategies.

Ünel et al. [UDUG04] proposed a query optimization on the basis of operator reordering. Here, a differentiation between internal node reordering and leaf node reordering is made. Similar to the approach introduced in this thesis, statistics about the retrieval environment are used to perform the reordering. A recent study of Wu et al. [WCW11] focused on already well-known relational operator reordering optimizations and their applicability to multimedia retrieval.

The work highlighted by Lin et al. [LC06] is focusing on the creation of a central index, that manages feature vectors for multimedia resources. Here, the retrieval problem is transformed into a string matching problem. To enable efficiency, two pruning techniques are in use. A similar approach is issued by Marin et

[95] This Chapter is partially based on [SSB+12].

[96] An comprehensive evaluation of the proposed optimizations is part of Chapter 12.

al. [MGCB08]. A distributed index data structure has been implemented following parallel computing techniques. This forms a hybrid index structure, which is a combination of the list of clusters and the sparse spatial selection indexing strategies. Both works are similar to the *semantic query cache* (see Section 10.3) proposed in this thesis.

A common optimization approach in relational database systems among different peers is an intelligent allocation of data. Manjarrez-Sanchez et al. [MSMV07] lifted these techniques to multimedia retrieval by performing a clustering on the data and allocating those clusters to various nodes. Obviously, manipulation as well as reorganization of data is not possible in the domain of an external metasearch engine.

10.2 Intra-Query Optimization

A federated query execution process can be split in several phases as shown in Section 8.7. The phases that directly affect an external metasearch engine are query decomposition (i), query segmentation (ii), global optimization (iii) and aggregation (iv) of the partial result sets. The remaining phases, e.g., local optimization (v) and local execution (vi), are implemented in the corresponding retrieval services. Both phases rely on expert knowledge of the underlying application domain as well as the exact data model. However, only the semantic links are exposed and the actual data model as well as its filling degrees of a particular backend are hidden to the external metasearch engine. Therefore, this thesis concentrates on phases i to iv.

Throughout phases i to iv, AIR is equipped with three major optimization strategies to enable an efficient multimedia retrieval, which will be highlighted in the following subsections.

10.2.1 Query Execution Planning

Query execution planning is one of the most important tasks in a (distributed) retrieval system. In general it specifies the execution strategy of query segments and is handled by the optimization component. The optimizer consists of three different parts: *search space*, *cost model* and the *search strategy*, which have been already introduced in Section 8.7.

Especially in the distributed domain, the calculation of costs and the selection relies on blurred values and statistics, such as the average or median response time of data stores. The presence of imprecise data makes it impossible to perform exact calculations. Further, an exact knowledge about internal characteristics of backends, e.g., item distribution, can not be given in a general case. Therefore, AIR uses heuristics to choose an query execution plan. In contrast to exact computations, heuristics do not guarantee the selection of an optimal solution for a problem. AIR is equipped with a buttom-up Greedy heuristic that performs the reordering of the operators on the basis of a certain weight that has been assigned to query types,

Table 10.1: Greedy operator reordering rules

Rule	Abstract rule	Tree type
α	$w_1 \approx w_2 \approx w_i \ll w_n$	left-deep tree
β	$w_1 \approx w_2 \approx w_i \approx w_n$	bushy tree

which are the leaves of the query tree. The formula is defined as follows and is a variant of Lohmann et. al. proposal:

$$w_{Total_i} = t_{processing_n} + \#result\_items_n, \tag{10.1}$$

with w_{Total_i} a weight assigned to a query type i, $t_{processing_n}$ the median processing time (including transfer and actual retrieval in the interpreter) and $\#result\_items_n$ the median number of result sets for a specific backend n. Here, $t_{processing_n}$ as well as $\#result\_items_n$ are calculated in the query cache component and are updated after each query processing. The weights are then propagated and combined through the query tree during reordering. Here, a high weight value means a longer predicted execution time of the query.

The Greedy algorithm includes two rules, performing the decision how the query tree shall be swapped as listed in Table 10.1. Those generate two different query trees with different behavior in its processing. Rule α constructs a left-deep query tree with *high weighted* query operators near to the root. This ensures that partial results of *lower weighted* query types can be already aggregated instead having the delay of waiting for the *heavier* ones. In contrast to that, rule β builds a bushy tree since all involved weights are within a certain interval. Here, the parallel distribution as well as execution can be fully exploited.

10.2.2 Query Processing Strategies

The query execution planning is one dimension of a query optimization technique. Another dimension is the actual strategy, how the subqueries will be executed and the partial result sets consolidated with respect to the overall query semantic. Here, two variants have been integrated, namely a *demand-driven* (Volcano model) and a *data-driven* (pipelining) aggregation strategy. The examples given in this section are based on the query specified in Section 8.7 of the isolated image retrieval use case.

Demand-driven processing The Volcano model [Gra94a] treats all operations of a query execution as iterators. It is demand-driven and therefore operator centric. Following this, they support a simple Open-Next-Close (ONC) protocol. The basic life cycle of a iterator is as follows: initialization, incrementation, termination of the loop due to a condition, and finalization. After the incrementation action, specific routines can be applied to a specific element, called support functions. The splitting

between iterators and support functions is crucial for independence of control and iteration. Further, it enables ease extension or modifications of functionalities. Iterators can be nested and then act like subroutines. In this light, the input section of an outer operator points to another query operation. The main design principle beyond the Volcano model is anonymous inputs, since a operator does not know where the input is from.

Figure 10.1 shows the demand-driven processing of the example query. A query execution chain is processed always in a bottom-down manner. Following this, an execution of open leads to a propagation of a call hierarchy leading to an initialization of all following iterators in a recursive way. Then, all data is allocated and the partial results are loaded. To generate the top most element, *next* is called. In all nested elements, next will be processed by the support function. The top most operation polls as long until it is able to produce one output element. *Close* defines the shut down of all iterators.

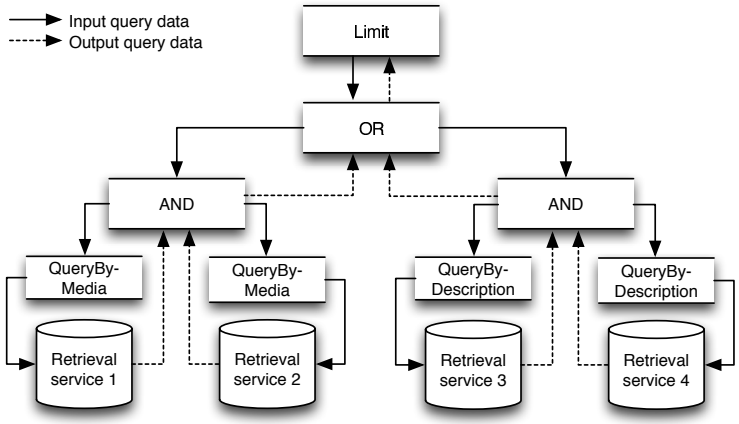


Figure 10.1: Demand-driven query processing with the Volcano model

Data-driven processing In relational database systems, pipelined query execution [LR05] is a common research topic. It offers a data-driven workflow and produces the complete result set with a single call and is therefore data centric. Within the aggregation phase, this model pushes the partial result item from the leafs towards the root node. To do this in a more efficient way, the tree is not fully executed per level. Beside from parallelism, I/O costs are the main issue of this model due to read and write operations. Obviously, the application of multiple operations to a set of partial result items before loading the next chunk of data reduces costs. If I/O is crucial together with CPU consumption, a data-driven approach is promising. Applying multiple operations to one result item in sequence grants a higher data locality in the CPU registers, respectively in the main memory. This reduces the overall I/O costs. In particular, this has a significant impact in the MEDICO use case scenario with respect to the processing of big multimedia data, e.g., CT scans.

For example, multiple projections and selections can be performed on a result item without the need to materialize it in any data structure of these operators. The properties of an operator define its ability, to be applied sequentially without materialization of the result item. The paths in the query execution tree, which are defined by sets of consecutive operators without breaking the data flow, are called *pipeline*. On the other hand, *pipeline breakers* are operators requiring more than one item to produce a result, e.g., the `Limit` operator. Figure 10.2 illustrates the above query example in a pipelined query execution. Here, two pipelines are generated.

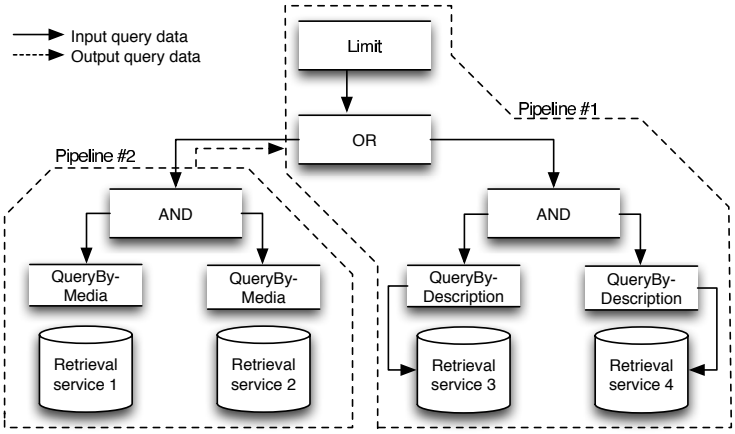


Figure 10.2: Workflow of pipelined query execution

Technically, the pipeline implementation integrated in AIR follows the well-known producer/consumer design pattern as proposed in [Neu11]. It is used to connect the individual pipelines accordingly to the query execution plan. Here, every pipeline must provide functionality to produce and consume data. To realize the data-driven approach, every pipeline is allowed to produce result items by calling the consume method of the parent pipeline. In order to avoid a pipeline from blocking during a produce call, every pipeline should have an input queue to consume data immediately.

Before executing a pipelined model, a conversion is necessary. Operators must be grouped into pipelines and mapped to threads. The implemented converter recursively traverses the input graph, starting with a provided entry node. While traversing the operator tree, the different types of the nodes are determined. This allows distinguishing between pipeline breakers and non-materializing operators. When a pipeline breaker is detected, the current pipeline is finalized and a new one is created. Breaking operators are wrapped and embedded in the pipelined operator tree. A pipeline can contain multiple operators that are executed in sequence. The content of a pipeline and the number of contained operators depends on the characteristic of the input query tree. It is important to decide if it is better to group many operators in small number of pipelines or to maximize the number of pipelines to improve parallelism. This decision depends on multiple factors, like

optimal number of threads and memory consumption per pipeline.

10.3 Inter-Query Optimization

The techniques presented in Section 10.2 are focusing on the processing of a single query. The inter-query optimization aims in the efficient processing of query streams by introducing a *query scheduler* and a specific *multimedia caching system*.

10.3.1 Query Scheduler

Evaluations showed, if many queries were active at the same time, AIR highlighted a deficit in the overall execution time due to high parallelism. The processing of an active query is internally split on several threads, leading to many concurrent context switches. These slowed down the processing time dramatically, because more time was spent on context switching than on result computing. In order to resolve this problem, a query scheduler was implemented restricting the upper bound of parallel executed queries. For a fair implementation, the First-Come First-Served (FCFS) paradigm is respected. The number of parallel queries is configurable to the amount of active CPU cores, as shown in the evaluation in Chapter 12. Remaining pending queries are blocked in a queue until a free slot is available.

10.3.2 Multimedia Caching System

A mediator framework, such as AIR, benefits from a cached query with a tremendous speed up due to a reduction of distribution of query segments to connected retrieval services. This means an improvement of the execution time up to the maximum round trip delay time. Additionally, a possibly expensive and time-consuming aggregation of various result sets can be omitted. The cache system not only considers the root node for caching, but also subtrees and leafs. Each cache hit in the query tree entails shorter processing time since partial results are already computed. Especially in case of scenarios enabling query refinement the caching of subtrees has an significant impact.

The cache system is integrated in the query cache and statistics component of AIR and can be further divided into three sub-processes following a top-down principle: *Calculate hash*, *cache search (look-up)* and *cache task*. Internally, it uses two different cache databases, which will be introduced next. The core feature of AIR is the distinction between regular textual query caching as well as a semantic multimedia caching. Both variants will be described next.

Textual query hashing. For a regular query, an unique identification for every tree node has to be computed. This approach uses the MD5 hash function. In order to remain general and open for extensions of new query conditions, the MPQF XML serialization of query conditions is used as an input for the hash function.

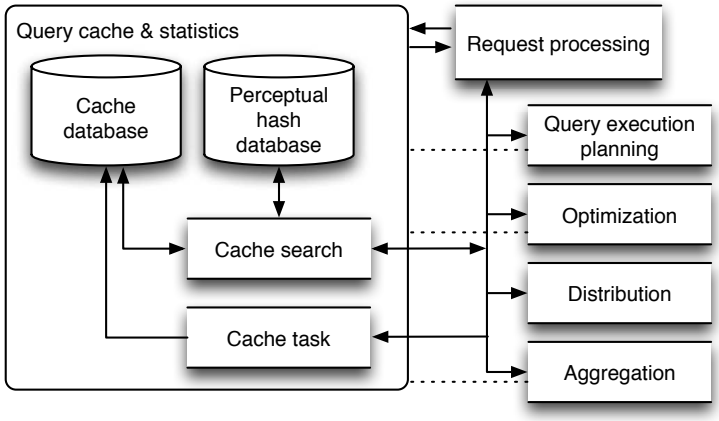


Figure 10.3: Architecture of the query cache and statistics component of AIR

In a preprocessing step, the XML serialization has to be cleaned to avoid irrelevant informations, which would pollute the hash function, e.g., XML comments. Identifications for boolean connectors and operators are calculated on a different way. They combine the hash values of the children and a small own identification to a unique representation of the node. Thus, hash values has to be computed bottom-up, because underlying hash values serve as input for nodes above.

Multimedia query hashing. A special focus lies on the caching of multimedia data encapsulated in specific MPQF query conditions, e.g., `QueryByMedia`. Normal hash procedures would not consider the perceptional content of multimedia data. For example, the same image could be specified with two different URIs. A hash function like MD5 would generate completely different hash values for equal images from humans point of view. On the other hand, a humans perception is robust against small variations of an image like color variations or cropping. AIR makes use of perceptual hash[97] [WP11] functions to enable semantic caching of multimedia data. The key idea is that different digital representations, which look the same to the human perception, should generate the same or a similar hash value. To be invariant against these kind of image changes, the perceptual hash functions extract certain significant features from the input multimedia object. Those features serve as an input for the hash functions and can be compared against each other by the Hamming distance.

In the context of multimedia data, the cache system should also produce a cache hit for similar multimedia objects. This can be achieved by the open source software Apache Lucene[98] serving as hash database. In general, Lucene is a high-performance, full-featured text search engine library. It is a technology suitable for nearly any application that requires full-text search and can be seen as de-facto

[97] AIR utilizes the open source library of `http://www.phash.org/`, last checked December 18, 2013.

[98] `http://lucene.apache.org/`, last checked December 18, 2013.

standard for search and indexing. In this work only a subset of its functionalities is used. Lucene can store a set of terms and offer the ability to search similar terms for a given input string efficiently. For the search functionality the known Levenshtein distance [Dam64] is used. Since all perceptual hash values have the same length, the computation of the Levenshtein distance is the same like the Hamming distance leading to the following workflow: First, a perceptual hash of the input multimedia resource is computed. Then, the Lucene database is queried for similar hash values. Therefore, a top-k query with $k = 1$ is send to Lucene. This means that only the best match is responded. Here, a minimum threshold of similarity is specified to ensure quality. If a similar hash value was found, it will be returned. If no corresponding was found, the original value is returned and stored in the database. As a result, similar multimedia resources can get the same unique identifier and therefore are regarded as the same in the following cache system.

Two-level caching architecture. The phases cache search and cache task are the same for both caching variants. Given a specific hash identifier for a execution plan node, the cache system can be searched for cached results. For efficiency reasons, this cache system is based on a two-level caching architecture. In fact, the two levels operate on different environments and storage layers. The first layer, called live nodes, operates in-memory and the second layer operates on a external database, called cache database. Figure 10.4 shows the complete workflow of the 2-level caching mechanism.

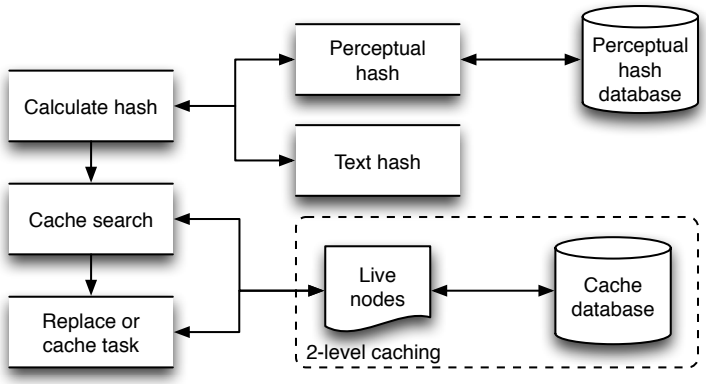


Figure 10.4: Workflow of the 2-level caching mechanism

The in memory solution represents a pool of all active nodes in AIR. These live nodes are shared across concurrent query processing threads. The concept is motivated by a query evaluation paradigm from relational databases called *on-demand simultaneous pipelining* [HSA05]. The advantage of this kind of cache layer is, that no new objects have to be instantiated and no memory space is wasted, because the nodes are already in the system. In order to enable the live node caching, two previously presented modifications were important. The cache system prunes the live node pool whenever a request processing is finished. No longer

required live nodes are removed and do not waste system resources.

The second layer is based on an external database, which actually stores the cached results. Motivated by the concept of cloud computing, a use case is conceivable that one AIR instance is no longer sufficient to perform a big quantity of simultaneously queries. Therefore the system can be load balanced with several distributed instances of AIR and a shared global cache system is required. For this reason the database approach for the second cache layer was selected, instead of a local approach. A key-value based record storage is sufficient to meet the requirements of the persistence cache layer. Hence, a NoSQL database was chosen, namely Apache CouchDB[99].

The result of a cache request affects the function of the cache task. If a cache hit occurs, the corresponding execution plan node or sub-tree is replaced with a special cached node including the precomputed results. Those behave like a normal execution plan node, therefore no modifications of the aggregation process is required. If a cache miss occurs, a cache task will be added to the execution plan. A cache task ensures that the results of this node will be stored in the cache system when results are available. The list of cache tasks will be processed after the distribution phase of the request processing. In fact, the implementation of cache tasks utilizes regular aggregation processes to avoid recomputing of results and waste resources.

[99] http://couchdb.apache.org/, last checked December 18, 2013

Chapter 11

Retrieval in Unfederated Multimedia Environments

The aforementioned inter-query optimization techniques for multimedia retrieval are only applicable if a homogeneous or heterogeneous retrieval environment is present, cp. Section 5.2. In terms of autonomous systems with no connections in the first place, further result fusion techniques have to be applied.

This chapter proposes a multimedia result fusion technique applicable in interoperable environments, completing the third contribution of this thesis. It presents an approach for a late result fusion performed inside an external multimedia metasearch engine. Its main innovations are the integration of Fuzzy logic to manage uncertainty present in similarity search, independence of a specific feature vector and the combination of rank and score information during the fusion process. The remainder of this chapter is organized as follows: Section 11.1 characterizes the issue of result fusion in the domain of external metasearch and related work is part of Section 11.2. Section 11.3 gives a formal definition of basic functions and the multimedia fusion operator whereas Section 11.4 investigates on the implementation of the algorithm and the integration into a external multimedia metasearch engine.

11.1 Characterizing the Issue

As already stated in Chapter 1, there exist a broad scope of multimedia sharing platforms. Most of them provide several ways of formulating a query, e.g., by an API or a query language, and offer diverse query functionalities, e.g., keyword-based search or query-by-example. Beyond the borders of formulating a unified query and addressing diverse metadata formats, a crucial task is to present a unified view of these different results sets. This chapter focuses on the already introduced notion of an external metasearch engine, potentially combining multiple evidences based on a variety of modalities, which (can) operate over different sets of documents.

Following this, we assume an external metasearch engine following a mediator architecture that is able to query diverse multimedia sharing platforms. Figure 11.1 shows a possible retrieval task in a generic retrieval environment. In this example, the query consists of a query image img_{qbm} combined with a set of semantic concepts $\{sc_1, \ldots, sc_z\}$, with $z \in \mathbb{N}$, describing the media resource. The query is segmented and transferred to different retrieval services, here RS_a and RS_b, both producing (possibly) different types of result sets.

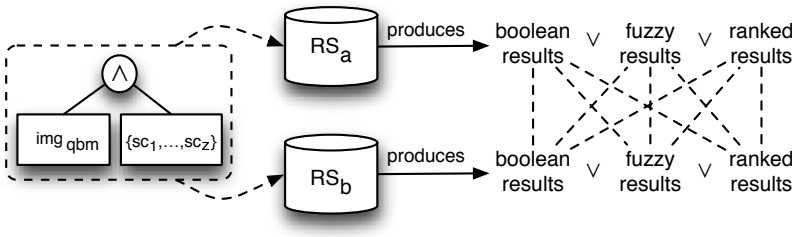


Figure 11.1: Case distinction of result set compositions

In this retrieval environment, three result set types can be identified for which the semantics are defined as follows:

i) Boolean results are reproducible results, valid to a certain condition (e.g., keyword match). These are mostly unordered sets of media resources.

ii) Fuzzy results indicate, that a reproducible result has been generated and the retrieved media resources are labeled with a distance value given by a fuzzy retrieval (e.g., similarity search). Here a normalized distance with range $[0 \ldots 1]$ is assumed. This value also implicates a (not necessarily unique) position in the overall result set, the rank value.

iii) Ranked results means that the result set is not reproducible. The result items are evaluated by a certain condition, cp. i), but labeled by a rank without evidence.

With respect to the quality of the overall result set, a mediator must be able to fuse each combination of the identified combination of characteristics.

11.2 Related Work

There is a lot of research working on the combination of result sets of multiple queries issued on the same or different document collections. Early work on the combination of retrieval results includes experiments from Fisher and Elchesen, who showed that retrieval results were improved by combining two Boolean searches over two different document representations [FE72]. In 1997, Lee presented his hypotheses on conditions for successful result fusion which initiated a number of contributions in the research community [LCS97]. Following Montague and Aslam, traditional fusion techniques can be divided into rank- and score based fusion methods [MA02].

Popular score-based methods include CombSum, (Weighted) CombMNZ, CombMin, CombMax, or COmbANZ which all combine multiple retrieval scores using different strategies by, for instance, summing up multiple retrieval scores or taking

the minimum, maximum, or average of the retrieval scores [SF94, LCS97]. Rank-based methods include rCombMNZ which uses ranks instead of relevance scores or ReciprocalRankFusion which sums up the reciprocal of the rank of each result [CCB09]. Voting-based methods were proposed by Montague and Aslam, first based on the Borda count, a positional voting algorithm, and later based on the Condorcet-fuse model, a majoritarian voting algorithm [MA02].

Furthermore, specialized score-based fusion strategies were proposed for multimedia retrieval which include the linear combination [YH03] or the min/max aggregation of scores [YHJ03]. Others applied machine learning techniques [YYH04, TN11] or optimization methods [WSF10, WK10] to determine the weights for various retrieval strategies.

In recent research efforts, the terms *early* and *late fusion* [SWS05, EHSM08, SGN+11, SLP11] are widely used in retrieval environments, that combine collections of media resources (possibly) from different modalities (e.g., visual or textual information). Both methods are using features describing a media resource for solving a pattern recognition problem. The difference between these methods lies in the process chain, whether the features will be combined as an input for the categorization (early fusion) or are used separately for categorization (late fusion). In the latter, the combination of the media resources will be performed on the results of the categorization process. For categorization tasks, most approaches use machine learning techniques, such as SVM, which are rather computationally intensive (e.g., learning phases). In contrast to that, the proposed approach follows the workflow of late fusion by using the more lightweight Fuzzy logic as reasoning technique to solve the categorization process.

11.3 Definitions and Notations

The outlined approach tackles the issue of a late result fusion strategy conducted inside an external multimedia metasearch engine improve merging of non-federated multimedia results. The key innovations of this proposal are:

- utilization of Fuzzy logic to handle uncertainty present in similarity search
- independence of a specific multimedia feature
- combination of rank and score information during the fusion process

In this section, the `FuzzyMultiMediaFusion` operator, Ξ, will be formally defined. It can be used with any feature vector describing a media resource. With respect to the *isolated image retrieval* application scenario, the overall description uses images as representatives for media resources without loss of generality.

In the following a distance metric *dm* is assumed to follow the well-known characteristics of a metric space, namely non-negative, identity of indiscernible, symmetry and triangle inequality. Further, the domain of a normed distance between

Table 11.1: Application of $NULL$ value to result set types

Result set type	Entropy of result set
Boolean results	$\{(img_1, \perp, \perp), \dots (img_M, \perp, \perp)\}$
Fuzzy results	$\{(img_1, dist_1, r_1), \dots img_M, dist_M, r_M)\}$
Ranked results	$\{(img_1, \perp, r_1), \dots img_M, \perp, r_M)\}$

two images img_i and img_j is defined as follows: $dist_{i,j} \in [0 \dots 1]$, with 0 as most similar and 1 indicating dissimilarity and $i, j \in \mathbb{N}$.

Let $A = \{(img_1, dist_1, r_1), \dots, (img_M, dist_M, r_M)\}$ be a result set of RS_a, with $|A| = M \wedge M \in \mathbb{N}$, img_i a result image, $dist_i$ a normed distance to the query image and r_i a given rank, representing the position of img_i in the result set. The result set $B = \{(img_1, dist_1, r_1), \dots, (img_N, dist_N, r_N)\}$, with $|B| = N \wedge N \in \mathbb{N}$, of RS_b, is defined analogous.

As stated beforehand, some combinations of different retrieval services produce result sets, where the score values and/or the ranks are unspecified leading to a diverse entropy on the instance level of the result sets. In order to fuse the available data, the missing information is modeled by a specific value, the $NULL$ value ($\perp$). Table 11.1 shows the application of the $NULL$ value to the identified result sets.

Based on this nomenclature, we define auxiliary functions and the operator. For the rest of the paper, a basic knowledge of Fuzzy logic, esp. Fuzzy inference [GNW95], is postulated. Central definitions can be found in Appendix A.

Definition 4 (*calcDist*)

The calcDist function calculates the normed distance, $dist_{i,qbm}$, between an image img_i and the query image img_{qbm} as well as to all images available in the second result set, here called neighborhood distance, $ndist_{i,1}$ to $ndist_{i,O}$. It is formally defined by:

$$\begin{gathered} calcDist_{f,dm}((img_i, dist_i, r_i), C) = \\ \{(dist_{i,qbm}, ndist_{i,1}, \dots, ndist_{i,k}, \dots, ndist_{i,O})_{img_i} | \\ dist_{i,qbm} = dm(f_{img_i}, f_{qbm}) \wedge \\ ndist_{i,k} = dm(f_{img_i}, f_{img_k})\} \end{gathered}$$

with f assumed a feature vector, dm a distance metric, $img_j \in C$, $1 < k \leq O$ and $(img_i \in A \wedge C = B \wedge O = N) \vee (img_i \in B \wedge C = A \wedge O = M)$.

The workflow of the *calcDist* function is illustrated in Figure 11.2. Beside the distance calculation to the query image img_{qbm} *calcDist* behaves like a Cartesian product.

Definition 5 (*makeFuzzyInf*)

The makeFuzzyInf function categorizes two images img_i and img_j into a similarity group, gr_{sim}, by utilizing the distance to the query image img_{qbm}. Further,

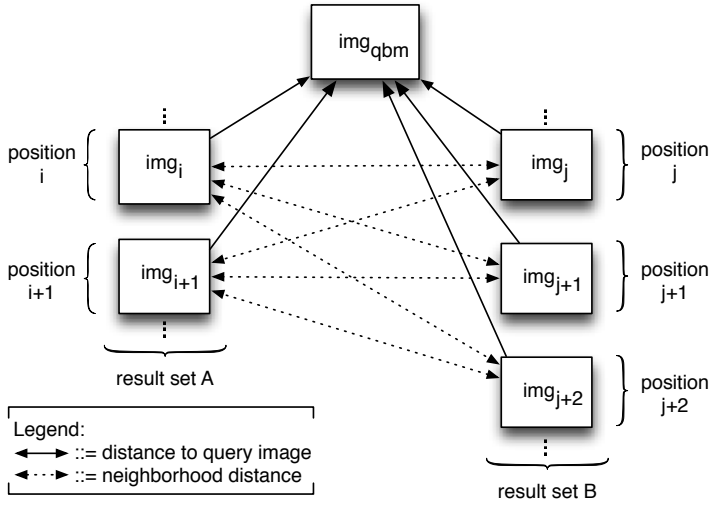


Figure 11.2: Illustration of the *calcDist* function workflow

makeFuzzyInf calculates a combined distance value for the image pair img_i *and* img_j, $dist_{fuz}$.

$$makeFuzzyInf((dist_{i,qbm}, ndist_{i,1}, \ldots, ndist_{i,N})_{img_i}, (dist_{j,qbm}, ndist_{1,j}, \ldots, ndist_{M,j})_{img_j})) = \Big\{(gr_{sim}, dist_{fuz})_{img_i,img_j} | fuzzyInf((dist_{i,qbm}, ndist_{i,1}, \ldots, ndist_{i,N})_{img_i}, (dist_{j,qbm}, ndist_{1,j}, \ldots, ndist_{M,j})_{img_j})\Big\},$$

where $img_i \in A \wedge img_j \in B \wedge 1 < i \leq N \wedge 1 < j \leq M$ *and* $fuzzyInf$ *be a Fuzzy inference.*

The groups gr_{sim} *are defined as three linguistic terms by triangular functions:*
i) "Very similar":

$$f_{sim}(dist_{i,qbm}) = \begin{cases} -2 * dist_{i,qbm} + 1 & , \text{if } 0 < dist_{i,qbm} \leq 0,5 \\ 0 & , \text{otherwise} \end{cases}$$

ii) "Related":

$$f_{rel}(dist_{i,qbm}) = 1 - 2 * |0,5 - dist_{i,qbm}|$$

iii) "Dissimilar":

$$f_{dsim}(dist_{i,qbm}) = \begin{cases} 0 & , \text{if } 0 < dist_{i,qbm} < 0,5 \\ 2 * dist_{i,qbm} - 1 & , \text{otherwise} \end{cases}$$

Table 11.2: Fuzzy associative matrix

x \ y	very similar	related	dissimilar
very similar	very similar	very similar	related
related	very similar	related	dissimilar
dissimilar	related	dissimilar	dissimilar

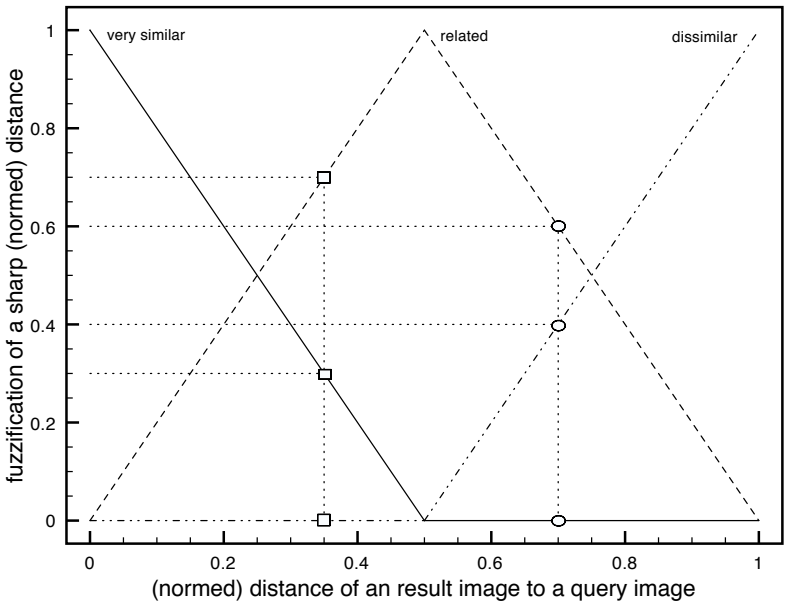


Figure 11.3: Fuzzification of a sharp (normed) distance

The basic steps and the configuration of Fuzzy inference are fuzzification (calculation of the degree of belonging to all linguistic terms), implication (Mamdani), inference (MAX\MIN) and defuzzification (first of maxima). The utilized Fuzzy associative matrix is given in Table 11.2.

A graphical representation of the triangular functions can be found in Figure 11.3. The definition of only three similarity groups is based on the fact that a distance $dist_{qbm}$ for an image img_i to the query image img_{qbm} will be lowered ("very similar"), increased ("dissimilar") or unmodified ("related").

Example Fuzzy inference. Let $dist_{1,qbm} = 0,35 (= x)$ be the distance of $img_1 \in A$ to img_{qbm} and $dist_{2,qbm} = 0,7 (= y)$ be the distance of $img_2 \in B$ to img_{qbm}. Figure 11.3 illustrates the fuzzification process for both distances, resulting in the degrees of belonging given in Table 11.3.

The associative matrix (see Table 11.2) defines nine different fuzzy rules of the following form: IF $x (\in gr_{sim})$ AND $y (\in gr_{sim})$ THEN $z (\in gr_{sim})$. The degrees of belonging will be inserted into the rule set to calculate a degree of fulfillment using

Table 11.3: Degrees of belonging of $dist_{1,qbm}$ and $dist_{2,qbm}$

	very similar	related	dissimilar
$dist_{1,qbm}(= x)$	$0,3$	$0,7$	0
$dist_{2,qbm}(= y)$	0	$0,6$	$0,4$

MIN (Mamdani) as fuzzy operator[100]:

$$\begin{aligned} MIN(\mu_{x=sim}; \mu_{y=rel}) &= MIN(0,3; 0,6) = 0,3 \\ MIN(\mu_{x=sim}; \mu_{y=diss}) &= MIN(0,3; 0,4) = 0,3 \\ MIN(\mu_{x=rel}; \mu_{y=rel}) &= MIN(0,7; 0,6) = 0,6 \\ MIN(\mu_{x=rel}; \mu_{y=diss}) &= MIN(0,7; 0,4) = 0,4 \end{aligned} \tag{11.1}$$

The MAX\MIN inference results in:

$$\begin{aligned} &MIN(0,3; \mu_{z=sim}) \\ &MIN(0,3; \mu_{z=rel}) \\ &MIN(0,6; \mu_{z=rel}) \\ &MIN(0,4; \mu_{z=diss}) \end{aligned} \tag{11.2}$$

From a graphical point of view, a degree of fulfillment cuts the corresponding triangular function of the linguistic term at its value.

Next, the first of maxima defuzzification utilizes the highest degree of fulfillment of the rules i to iv (here rule iii with $0,6$) and takes the most left of this maxima for the corresponding linguistic term (here *related*). This linguistic term defines the similarity group gr_{sim} for the image pair img_1 and img_2. The combined distance value is calculated by solving the triangular function defined by the maxima, here f_{rel} with the degree of fulfillment, here $f_{rel} = 0,6$. The function can be solved by $x = 0,3$ or $x = 0,7$, both spanning an interval at the x-axis describing all possible values for $dist_{fuz}$. Following left of maxima, $dist_{fuz} = 0,3$ for the image pair img_1 and img_2.

The outcome of the Fuzzy inference will be directly used by the `boostIt` function. In principle this function is in charge of combining of combining the calculated distances into shift vales. It is formally defined as follows:

Definition 6 (*boostIt*)

The boostIt function calculates a set of shift values, b_1 to b_O, for an image img_i utilizing its relation to the neighbors in the corresponding result set. In the following, let $C_{img_i,img_j} = \{(img_i, dist_i, r_i), (dist_{i,qbm}, ndist_{i,1}, \ldots, ndist_{i,N})_{img_i}, (gr_{sim}, dist_{fuz})_{img_i,img_j})\}$ be a set of all calculated values related to an image

[100] *Note:* Rules equal to zero are not listed.

img_i. boostIt is calculated as follows:

$$boostIt_{img_i}(C_{img_i,img_j}) = \{(b_1,\ldots,b_k,\ldots,b_O)_{img_i} | f_{boost_k}(C_{img_i,img_j})\},$$

$$\text{and } f_{boost_k}(C_{img_i,img_j}) = \begin{cases} -\dfrac{dist_{fuz} + ndist_{i,j}}{2 * r_i} * w_b & \text{, if } gr_{sim} = \text{“very similar”} \\ \dfrac{ndist_{i,j}}{r_i} & \text{, if } gr_{sim} = \text{“related”} \\ \dfrac{dist_{fuz} + ndist_{i,j}}{2 * r_i} * w_b & \text{, if } gr_{sim} = \text{“dissimilar”}, \end{cases}$$

with w_b a configurable weighting factor for $(1 < k \leq O = N \wedge img_i \in A \wedge img_j \in B\ ) \vee (1 < k \leq O = M \wedge img_i \in B \wedge img_j \in A)$.

A single shift value b_i indicates the correspondence to the linguistic term. If the algebraic sign is negative, it has a positive effect on the similarity and vice versa for positive algebraic signs. A shift value b_i also takes the rank r_i of an result item into account by its multiplicative inverse. This means, the higher its position in the result set, the lower its effect in the calculation.

The consolidation of the shift values is part of the `calcSim` function. Here, the initial distance of an image to the query in the overall result set will be recomputed with respect to the shift values.

Definition 7 (*calcSim*)

The calcSim function is calculating a new score value $dist'_i$ for an image img_i by the use of the standard deviation of the shift values $(b_1,\ldots,b_O)_{img_i}$. It is defined as follows:

$$calcSim_{img_i}((C_{img_i,img_j}),(b_1,\ldots,b_O)_{img_i}) = \{(img_i, dist'_i, r_i) | dist'_i = dist_{i,qbm} + \sigma_{(b_1,\ldots,b_O)_{img_i}}\},$$

with $\sigma_{(b_1,\ldots,b_O)_{img_i}}$ the standard deviation of the sample, here $(b_1,\ldots,b_O)$ and $img_i \in A \cup B$.

On the basis of the new distance value, an altering of the actual position in the result set can be given. This will be conducted by the `calcRank` function.

Definition 8 (*calcRank*)

The calcRank function is a total order over a set of items defined by the less-than-or-equal relation, $\leq$.

$$calcRank(A \cup B, \leq) = \{\forall (img_i, dist'_i, r_i), (img_j, dist'_j, r_j) \in A \cup B : dist'_i \leq dist'_j \vee dist'_j \leq dist'_i\}$$

The last building block is the `FuzzyMultiMediaFusion` operator itself, integrating `calcSim` and `calcRank`.

Definition 9 (*FuzzyMultiMediaFusion*)

The binary Fuzzy multimedia fusion operator, Ξ, combines the result sets A and B. It is formally defined as follows:

$$A \Xi B = \Big\{ (img_i, dist'_i, r'_i) | (img_i, dist_i, r_i) \in A \cup B \ \wedge$$
$$dist'_i = calcSim_{img_i}((img_i, dist_i, r_i), (b_1, \ldots, b_O)_{img_i}) \Big\},$$

where $calcRank(A \cup B, \leq)$ *applies and defines the new rank value* r' *and* $1 < i \leq |A| + |B|$.

11.4 Algorithm Inspection & System Integration

The formal definitions of Section 11.3 serve as a basis for the actual implementation of the FuzzyMultiMediaFusion. Algorithm 1 illustrates the arrangement of the defined functions and the overall workflow in pseudo code. Internally, *MultiDistanceItem* (MDI) is used to store calculation results for a specific image img_i. In particular, it holds the following values: extracted feature vector, query image distances, neighborhood distances, fuzzification results and shift values.

Data: A, B
Result: C_{fused}
1 $mdiList.init()$;
2 $C_{fused}.init()$;
3 **foreach** $(img_i, dist_i, r_i) \in A \cup B$ **do**
4 $\quad MDI_{img_i} \leftarrow calcDist_{f,dm}(img_i, dist_i, r_i)$;
5 $\quad mdiList.add(MDI_{img_i})$;
6 **end**
7 **foreach** $(img_i, dist_i, r_i) \in A$ **do**
8 $\quad$ **foreach** $(img_j, dist_j, r_j) \in B$ **do**
9 $\quad\quad MDI_{img_i} \leftarrow mdiList.get(MDI_{img_i})$;
10 $\quad\quad MDI_{img_j} \leftarrow mdiList.get(MDI_{img_j})$;
11 $\quad\quad makeFuzzyInf(MDI_{img_i}, MDI_{img_j})$;
12 $\quad$ **end**
13 **end**
14 **foreach** $MDI_{img_i} \in mdiList$ **do**
15 $\quad boostIt_{img_i}(MDI_{img_i})$;
16 $\quad (img_i, dist'_i, r_i) \leftarrow calcSim_{img_i}(MDI_{img_i})$;
17 $\quad C_{fuse}.add(img_i, dist'_i, r_i)$;
18 $\quad mdiList.remove(MDI_{img_i})$;
19 **end**
20 $sort(C_{fuse})$ according to $calcRank()$ ascending;
21 **foreach** $(img_i, dist'_i, r_i) \in C_{fuse}$ **do**
22 $\quad (img_i, dist'_i, r'_i) \leftarrow C_{fuse}.getPosition(img_i)$;
23 **end**

Algorithm 1: FuzzyMultiMediaFusion()

The calculation of $makeFuzzyInf$ (line 7 to 13) is symmetric and the results are directly stored in the MDIs. Some processing steps of the proposed algorithm can be performed in parallel by a configurable amount of threads. These are the feature extraction (lines 3 to 6) and the calculation of the shift values as well as the calculation of the new distance (line 14 to 19).

The proposed approach will be made available inside AIR with a set of algorithms serving as physical implementation of the AND operator. It will be injected in the query execution, if retrieval task addressing an unfederated multimedia environments is present, as shown in Figure 11.1. To be compliant with the already implemented strategies of the mediator for an AND operator (e.g., Nested Loop Join), the FuzzyMultiMediaFusion implementation also follows the Volcano model (Open-Next-Close) [Gra94b]. This architecture along with the mathematical constraints of the approach (e.g., t-norm of Fuzzy logic) enables operator reordering and therefore optimization in the query execution planning phase. All components are written entirely in Java.

Chapter 12

Evaluation

To ensure validity of the proposed techniques, a prototypical implementation of the AIR framework equipped with the optimization techniques has been implemented in Java.

This chapter evaluates critical parts of the AIR framework. While Section 12.1 defines the quality metrics used during the evaluations, the remaining chapter is subdivided in two parts consequently: The first part (Section 12.2) covers the evaluation of the optimization techniques for federated multimedia retrieval. Besides the description of the evaluation environment (see Section 12.2.1) it forks into the comparison of intra-query optimization strategies (Section 12.2.2) and the results of the inter-query optimization (Section 12.2.3). The second part (Section 12.3) analyses the practical retrieval abilities of the Late Fuzzy Fusion operator. Here, the evaluation environment and the algorithm setup are given in Section 12.3.1. The actual evaluation is split in a benchmarking-based (Section 12.3.2) and user-centric evaluation (Section 12.3.3). Concluding remarks of the evaluation are discussed in Chapter 13.

12.1 Quality Measures

In general, an (multimedia) information retrieval system performs a categorization of a given set of (media) resources in two groups while evaluating a specific query condition: *relevant* or *non-relevant*. The resources marked with relevant are finally stored in the result set and are *retrieved* by the system. Non-relevant resources are discarded. To evaluate the overall retrieval quality of such a system, the research community proposed several quality measures. The most well known measures for such a task are *precision*, *recall* and *f-measure*. In the following their definitions will be given on the basis of [MRS08] and their correlations are discussed.

Definition 10 (*Precision*)

The set of retrieved documents that are relevant to the search specifies the precision value. It is calculated as follows:

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Definition 11 (*Recall*)

The fraction of the documents that are relevant to the query and are successfully retrieved define the recall value. It is calculated as follows:

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

Obviously, an information retrieval system cannot optimize both values. An analysis of the application scenario indicates, whether precision or recall should be prioritized. An example shows this correlation: the highest recall value of `1` can be simply reached in *each* information retrieval system by retrieving each resource in the collection for any query. In this case, all relevant resources are present in the result set (not taking into account their positioning). On the other hand, non-relevant resources in the result set apparently minimize the precision value. A contingency matrix as illustrated in Table 12.1 is used to illustrate these observations. Further, the matrix shows the origin of the well known terms *true / false positives* and *false / true negatives*.

Table 12.1: Contingency matrix for retrieved resources

	Relevant	Non-relevant
Retrieved	true positives	false positives
Not retrieved	false negatives	true negatives

A variation of precision and recall is its limitations to the top-k result set items. Those variants are heavily in use in the web retrieval domain and are marked with *precision@k* as well as *recall@k*. The research community called for a single measure that combines precision and recall, which led to the definition of the *f-measure*. It is formally defined as follows:

Definition 12 (*F-measure*)

The f-measure is the weighted harmonic mean of precision and recall:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R},$$

where $\beta^2 = \dfrac{1-\alpha}{\alpha}$, $\alpha \in [0,1]$ *and* $\beta^2 = [0,\infty]$.

Typically, *default balanced f-measure* is in use, which equally weights precision and recall. In this case, $\alpha = 1/2$ or $\beta = 1$ is applied.

It is clear that the aforementioned quality measures evaluate the retrieval ability only with a minor focus on the actual ranking of the result items. For this task, the *discounted cumulative gain (DCG)* and the *normalized discounted cumulative gain (nDCG)* [JK02] have been proposed. Here, the DCG indicates the relevance of the result items, whereas the nDCG measures the arrangement/ranking of the result items. It is based on the observation, that a highly relevant resource is more useful, if its rank is higher compared to non-relevant resources. Both, DCG and nDCG,

are calculated with respect to the top-k search results. They are formally defined as follows:

Definition 13 (*Discounted Cumulative Gain (DCG)*)

The DCG penalizes relevant resources that occur lower in the overall rank logarithmically proportional to its position. It is calculated for a particular rank position p and is defined as:

$$DCG_p = rlv_1 + \sum_{i=2}^{p} \frac{rlv_i}{\log_2(i)},$$

where rlv_i is the relevance of the resource at position i in the result set A and $i \in [0, \ldots, |A|]$.

There exist variations of the classical DCG definitions by exchanging the logarithmic reduction factor by other mathematical functions. Nevertheless, recent research results [WWL+13] showed, that the logarithmic reduction factor is recommended best practice.

Definition 14 (*normalized DCG (nDCG)*)

The normalized DCG indicates a normalized measure to indicate an average quality estimation of a search engine. Following the DCG, it is calculated for a particular rank position p and is defined with:

$$nDCG_p = \frac{DCG_p}{IDCG_p},$$

with IDCG the ideal reachable DCG value for the current position p.

Example. An example calculation of the DCG/nDCG will be given next: Let $A = \{res_1, res_2, res_3, res_4, res_5\}$ be a result set of resources res_i, with $i \in \{1, 2, 3, 4, 5\}$, of a query q produced by the retrieval service s. A user judges the ranking of the result set with the following relevance scores: 0 the position of the element is non-relevant, 1 the position of the resource is neutral and 2 the position of the resource is acceptable. For this example, we assume the following applied relevance scores added to the rankings of the resources:

$$res_1 \to 2,\ res_2 \to 0,\ res_3 \to 1,\ res_4 \to 2,\ res_5 \to 0$$

Following this, the calculation of DCG can be conducted:

$$DCG_5 = rlv_1 + \sum_{i=2}^{5} \frac{rlv_i}{\log_2(i)} = 2 + (0 + 0,63 + 1 + 0) = 3,63$$

Next, the user rating sorted by its relevance scores specifies the IDCG value:

$$res_1 \to 2,\ res_4 \to 2,\ res_3 \to 1,\ res_2 \to 0,\ res_5 \to 0$$

$$IDCG_5 = rlv_1 + \sum_{i=2}^{5} \frac{rlv_i}{\log_2(i)} = 2 + (2 + 0,63 + 0 + 0) = 4,63$$

Finally, the nDCG can be computed:

$$nDCG_5 = \frac{DCG_5}{IDCG_5} = \frac{3,63}{4,63} = 0,78$$

12.2 Evaluation of Optimization Techniques

The first part of the evaluation focuses on the implemented optimization techniques in terms of federated multimedia retrieval[101].

12.2.1 Evaluation Environment

The evaluation of the optimization strategies have been conducted in a special environment, since the real world use case of Section 8.2 may introduce unwanted latencies.

From a technical point of view both evaluations have been conducted on an Intel Core 2 Due notebook with a 2.0 GHz processor. The Java Virtual Machine x64 (JVM) was running on a Windows 7 x64 operating system. To be able to deliver meaningful test results, large result sets were needed. Therefore, the three gigabytes of memory with an upper bound of four gigabytes were assigned to the JVM.

For both of the upcoming two evaluation sections, the same backends are used, but the enclosed evaluation data is different. In general, the benchmarking suite in use imitates a federated, interoperable image search.

Details on data and query setup of query execution strategies. Section 12.2.2 highlights the difference between demand-driven and data-driven query processing strategies. The connected retrieval services consist of dummy backends that answer with a configurable portion of data. The decision to use artificial result sets is based on the fact to easily scale up and to have a proper alignment of the data. The connected backends emulate a similarity as well as a metadata-based search (details in the according appendices). All query tree leaves are retrieved result sets based on metadata-based queries and are therefore perfect for aggregation. Every result item has a unique identifier as well as five different and uniformly distributed descriptions, which are used for grouping. To test the join algorithms (AND), different but not disjunct result sets were used. Every result item will always find a join partner, what means that the final result set contains exactly the same quantity of items like a leaf. Additionally, every result set is randomly sorted with respect to the identifier. For this consideration, the following queries are in use:

- *Query I:* A single query leaf.

[101] This Chapter is partially based on [SSB+12].

- *Query II:* A *SortBy* (pipeline breaker) node with one leaf.
- *Query III:* A *Distinct* node with an appended *Projection*, which has again one appended leaf.
- *Query IV:* Ten leaves with nine *AND* operators above, forming a bushy tree.
- *Query V:* Ten leaves with nine *AND* operators above, each *AND* operator has exactly one leaf, forming a left deep tree.

Details data and query setup of query cache evaluation. The major goal of Section 12.2.2 is to compare the processing times of AIR with and without activated multimedia cache system. Hence, a very high cache hit rate is desired to make a statement about quality of the optimization. Additionally a good configuration for the query scheduler should be determined. In this part of the evaluation, queries combining metadata-based search with keywords and similarity search with example images are fired against the AIR framework. The connected retrieval services have a common data basis, which consists of 12164 images from the following downloaded ImageNet[102] synsets: *contact sport*, *ducks*, *castle*, *building*, *resort*, *shirt*, *pants*, *suv*, *sports car* and *flower*. These synsets are partly semantically correlated. The implementation of the metadata-based query service considers these semantic relations by a keyword search. Hence, a search for the keyword *car* would result in a union of image sets *SUV* and *sports car*. The content-based image search is implemented utilizing the *Lucene Image Retrieval library (LIRE)*[103]. The queries and their correlations are illustrated in Appendix C.1.

12.2.2 Comparison of Intra-Query Optimization Strategies

The first part of the evaluation is focuses on the comparison of demand-driven (ONC) and data-driven retrieval (pipelined). In all plots, the y-axis denotes the runtime in milliseconds that was needed to evaluate a query. The apparent runtime is the average of 25 executions for each measurement. The x-axis denotes the number of result items in every leaf of the query tree. Appendix C.2 contains boxplots to give an impression of the statistical distributions of runtimes for all queries. If there are ten leaves with the maximal quantity of 100.000 results items, the total quantity adds up to one million result items that must be processed.

The plot of query I in Figure 12.1 indicates a linear increase of runtime with the number of initial results. Checking result sets for duplicates is done in linear time, due to a hash table based duplicate check. Loading result sets in linear time is therefore possible. This is important to note, because leaves are part of every other query and for this reason add to their runtime. ONC and pipelining show almost identical values in this simple case.

[102] http://www.image-net.org/, last checked December 18, 2013.
[103] http://www.semanticmetadata.net/lire/, last checked December 18, 2013.

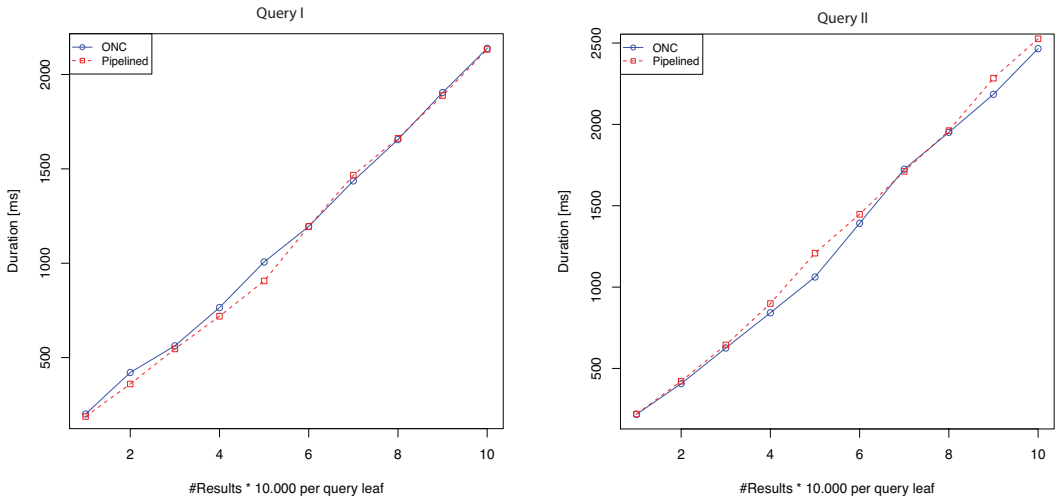


Figure 12.1: Evaluation of query processing strategies: Query I & II

The occurrence of a pipeline breaker is observable in the plot of query II in Figure 12.1. The implementation of the *SortBy* operator is largely based on the sort method provided by the Java standard library reaching a sorting complexity of $O(n * log(n))$. Since only one pipeline is generated, the *SortBy* operator is the bottleneck in the pipelining model and does therefore not outperform the ONC based approach.

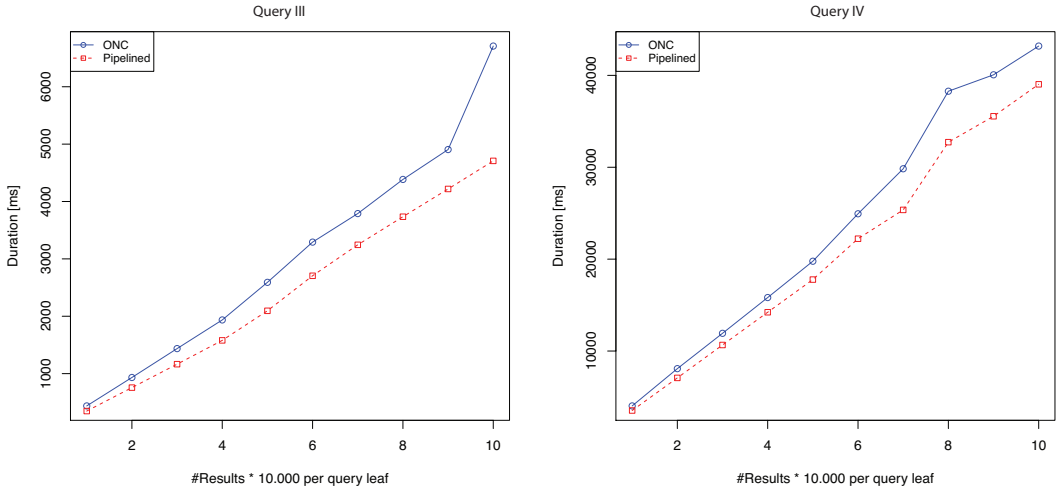


Figure 12.2: Evaluation of query processing strategies: Query III & IV

In contrast to the consideration of pipeline breakers, projection and distinct operators have to be evaluated due to regular occurrence in real world queries. The plot of query III in Figure 12.2 clearly shows that projection does not significantly add up to the runtime, because it is performed by a simple *SET* call. By using a

set data structure instead of a list structure the performance was be significantly improved. In this case, the pipelined model clearly outperforms the ONC model. Multi-threading for different data structures are not the reason for this, because they are equivalent in both models. Therefore, reduced materialization costs trigger this disparity.

The plot of query IV in Figure 12.2 shows the evaluation of a large query tree. Up to one million result items are loaded to memory, during evaluation of this large query tree. This causes a significantly increasing garbage collection overhead, when memory must be freed. Both figures indicate this at 80000 result items per leaf with a constant offset. The computation itself is still linear. Pipelined execution outperforms ONC based execution. The difference is clearly perceptible, but not as distinct as expected by using twice as many cores.

The same observation holds for the execution times of query V in Figure 12.3.

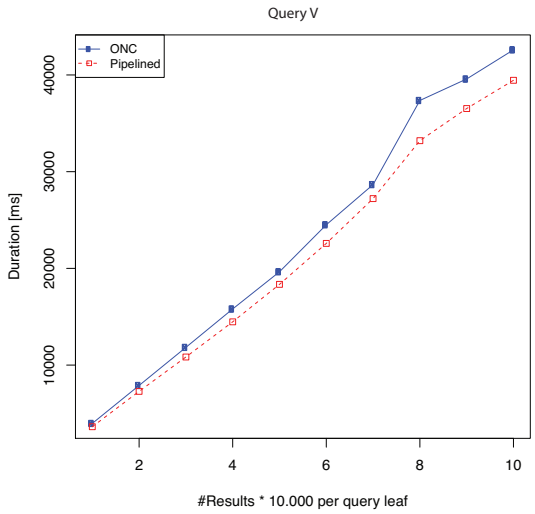


Figure 12.3: Evaluation of query processing strategies: Query V

12.2.3 Results of Inter-Query Optimization

The second part of the evaluation is analysing the performance of the multimedia query cache. A test run consists of 1001 queries, which are fired in parallel against the AIR framework. As explained beforehand, different combinations of image and keyword query types are used in the queries. Overall, a cach-hit rate of about 80% was reached, which could be split in 70% cache database and 10% live node access. In the following, different comparisons about request processing times will be given.

First, the request processing times[104] are compared. To get an impression of the system behaviour, the median request processing times for different combinations of query limiter and query transmission intervals have been computed. These results

[104]The processing times include time for the hole request, not only internal processing.

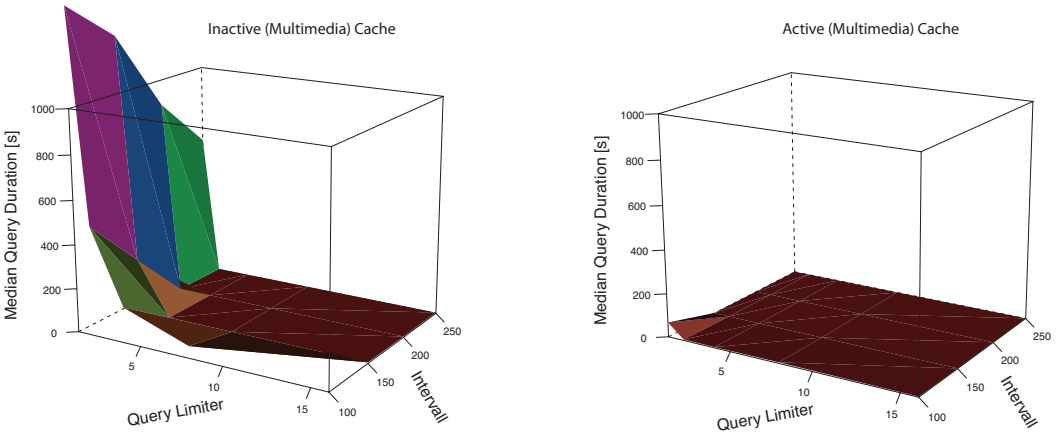


Figure 12.4: Evaluation of the multimedia query cache with variable query limiter and load

are visualized in Figure 12.4 without cache (left) and with active cache (right). The numerical execution times of these plots can be found in Table C.1 in Appendix C.1. It is observable that there are serious outliers in terms of the query processing times when the AIR framework is under a heavy load without caching mechanisms. In contrast to that, the system performs well with activated cache in these situations (small interval and few query limiters). As estimated, the diagram indicates a general decrease of processing time for all setting variations (average improvement is about 88%). Note that theses results are based on a very high cache hit rate. The median query duration is acceptable in almost all caching cases, except the setting with only one query limiter and the smallest query interval. Of course, this case (only one query can be active) is inconceivable in practice, but had to be taken into account for comparison. In the above defined hardware environment, the results show that a query limiter with eight active queries suits best for the case that caching is deactivated. A smaller count of limiter restricts the performance and a higher count slows down the system with to many context switches. In contrast to that, the system with activated caching can benefit from more parallelism because the system is under a smaller load.

In addition to the 3D-plots, two detailed box plots of Figure 12.5 enable a fine-granular view with respect to the actual query execution times as an example for all other queries and combinations. The properties of box plots[105] used in this thesis are as follows: The upper and lower boundaries depict maximum and minimum. The rectangle box contains 50% of all values (upper and lower quartile) and the median is marked with a black dash within this box. As a configuration, the query limiter is set to eight and the transmission interval to 200 ms.

Within the box plots, a significant improvement for all previously presented queries is observable. In this run, an average of 76% better processing times were

[105] This explanation of a box plot holds for the remaining thesis as well.

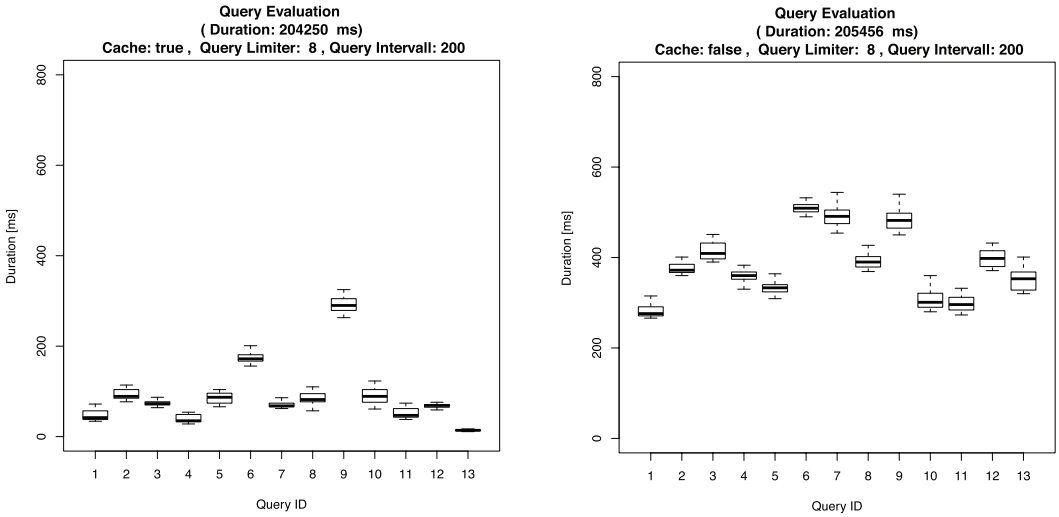


Figure 12.5: Evaluation of the multimedia query cache with fixed query limiter and load

measured with enabled cashing. Besides the evident fact of overall query optimization, it is noticeable that mostly all boxes shrink in the caching case, with query 13 as example. A smaller upper and lower quartile indicates that the variance decreases and therefore query processing time gets more regular.

12.3 Evaluation of the Late Fuzzy Multimedia Fusion

After the evaluation of the federated retrieval abilities of the AIR framework, this section concentrates on the retrieval abilities of the Late Fuzzy Multimedia Fusion.

12.3.1 Evaluation Environment and Algorithm Setup

The evaluation of the Late Fuzzy Multimedia Fusion will be settled in the overall application scenario of an interoperable image retrieval with the condition all retrieval services are isolated without well-known links between them. The test environment consists of a MacBook Pro equipped with an 2.7 GHz Intel Core i7, 4GB 1333 MHz DDR3 and 256 GB SSD. All images used in the evaluation offer a dimension of 500px fitted to long side equalling Flickr medium quality. The evaluation partners of the Late Fuzzy Multimedia Fusion (FMMF) are Round Robin (RR) [BBP03] and a simple (linear) multimedia fusion (SMMF). Round Robin simply removes the first element of a partial result, adds it to the final result list and moves on to the next partial result set, as long as items are present. In contrast to this non content-based fusion strategy, the simple multimedia fusion uses content information by extracting feature vectors from all available media resources. For the simple multimedia

fusion, a kNN-based implementation of the OpenIMAJ project[106] [HSD11] is in use. The query structure introduced in Section 11.1 (combination of keyword bases search and similarity search) will serve as an example to measure the quality of the three fusion strategies.

As the definition of the approach indicates, it is possible to tailor the algorithm to specific user needs by the configuration of various components. In order to select a specific multimedia feature for the configuration, various low-level image features have been evaluated. Here, a trade off between description quality and average extraction time is important, since the extraction is performed inside the mediator, which is inline with the definition of an external metasearch engine. The following feature algorithms[107] have been tested: colourLayout (CL), colourMoments (CM), CoocurrenceMatrix (CoMa), EdgeHistogram (EH), GlobalcolourHistogram (GCH), LocalcolourHistogram (LCH), Scalablecolour (SC) and SURF. Figure 12.6 shows the average feature extraction time per image in milliseconds.

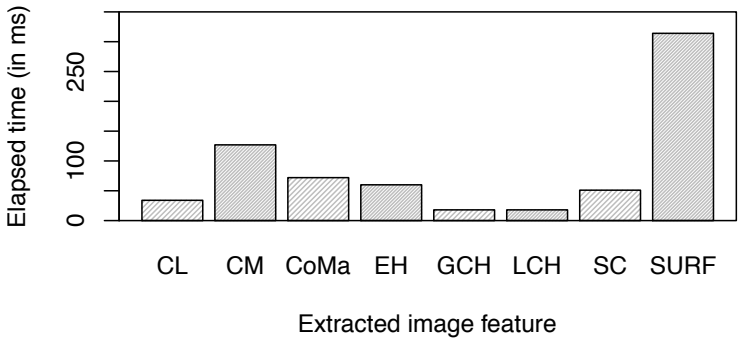


Figure 12.6: Average feature extraction time per image

SURF belongs to the SIFT family and is a sophisticated and robust feature. Unfortunately, it takes more than 300 ms per image for extraction. On the other hand, histogram based features (GCH&LCH) are calculated fast, but lack in robustness. For this evaluation, the CL feature of the MPEG-7 standard has been selected due to acceptable speed (30 ms avg.) and quality. The Euclidean distance is used as distance metric. Experiments have shown that a weighting factor $w_b = 0.25$ and 16 parallel threads (for this environment) are suitable.

Benchmarking-based evaluation data. In [ZBB+12] Zellhöfer observes that the judgment of a multimedia retrieval system is a cumbersome task due to the omnipresent imprecision, subjectivity and vagueness. Further, meaningful multimedia corpora strongly taking similarity queries into account are an on-going research topic and no direct suitable benchmark has been issued while time of writing. In

[106] http://www.openimaj.org/, last checked December 18, 2013.

[107] Used implementations of CL, EH and SC are the MPEG-7 reference implementation, CoMa, GCH and LCH have been implemented by the School of Information Technology, University of Sydney, and SURF is taken from the *ImageJ SURF* project.

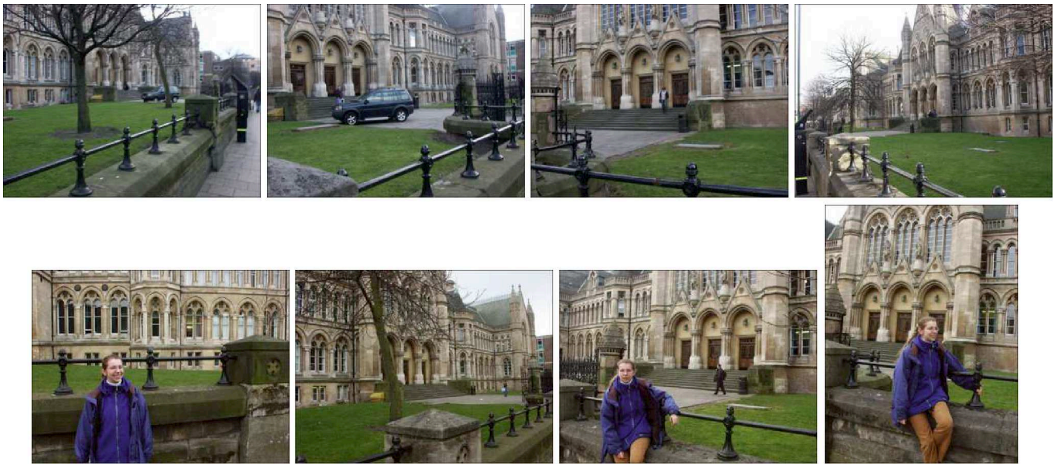


Figure 12.7: UCID: Selection of available images in the data set

Figure 12.8: Wang: Selection of available images in the data set

order to conduct well-grounded evaluations, Zellhöfer suggests using collections of media resources, in which each media resource is tagged with semantic concepts that are communicated by the media resource.

These findings have been taken into account for evaluating the merging abilities of the Late Fuzzy Multimedia Fusion as well as the two introduced evaluation partners. To ensure a standardized evaluation procedure, the `trec_eval`[108] [VH98] tool, which is also in use during the TREC conference series, is selected. This program calculates quality measures on the basis of a well-defined ground truth and the given results of an evaluation. To enable the calculation, it needs two files as input: a *qrel file* holding the ground truth by specifying the relevance of a media resource to a given topic and the actual results of the three result fusion algorithms stored in a *treceval file*.

As meaningful data basis, the following two by the multimedia community accepted image collections are used within the evaluation:

Uncompressed Colour Image Database (UCID). The main focus during the

[108] `http://trec.nist.gov/trec_eval/`, last checked December 18, 2013.

compilation of UCID[109] [SS04a] was to create a benchmark data set suitable for the quality judgement of compression algorithms since the images are made available in an uncompressed format. UCID consists of 1338 uncompressed images in the TIFF format that is tagged with 262 topics. In the average case, a topic contains about ten images. Besides, the data set also comes with a predefined ground truth to judge multimedia retrieval systems. For this evaluation, the images have been converted into the JPEG format. A selection of UCID images is illustrated in Figure 12.7.

Wang. The Wang image collection[110] [LW03] is a subset of the well-known COREL database [111] containing ten topics whereas each topic contains exactly 100 images. This results in an overall amount of 1000 images. Within this data set, the condition is given, that an image is only associated with one topic. A selection of Wang images is illustrated in Figure 12.8.

For both data sets, Zellhöfer provided the essential ground truth in the form of qrel files. In order to emulate an interoperable image retrieval system, queries combining metadata-based search with keywords and similarity search with example images are fired against the AIR framework. The retrieval services connected to the AIR framework are equivalent to those specified in the query cache evaluation environment of Section 12.2.1.

User-centric evaluation data. Besides the benchmarking-based evaluation, a qualitative evaluation on the basis of a real user evaluation has been conducted to investigate the impression of users in terms of the retrieval abilities. A real world test environment has been created, constituting of a LIRE instance for similarity search (fuzzy results) and a MPQF interpreter encapsulating Flickr for answering metadata-based query requests. The LIRE instance is filled with random pictures as well as semantically controlled data sets aligned to the evaluation queries of ImageNet resulting in an overall amount of approximately 20000 images. Both data stores are connected to the AIR framework. For this evaluation, two different queries are considered: Query a consists of a close-up image of a strawberry (red is dominant colour) as query-by-example combined by an AND with a metadata-based query containing the semantic concepts {strawberry, closeup} as keywords. In contrast to that, query b utilizes an image of a beach in Bali (uniform colour distribution) combined with the semantic concepts {Bali, beach} following the same structure as query a. The four semantic concepts also define the synsets for the crawled images of ImageNet. The queries are illustrated in Appendix C.3. The used quality measures in this part of the evaluation is DCG and nDCG.

[109] http://homepages.lboro.ac.uk/~cogs/datasets/ucid/ucid.html, last checked December 18, 2013.
[110] http://wang.ist.psu.edu/docs/related/, last checked December 18, 2013.
[111] https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval, last checked December 18, 2013.

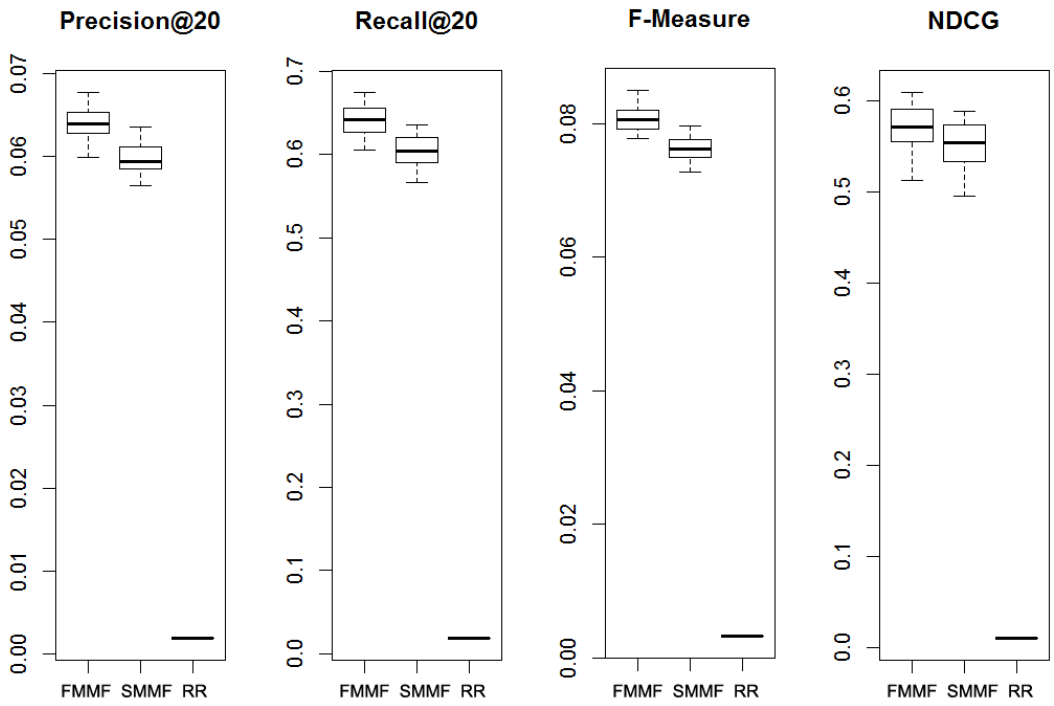


Figure 12.9: Evaluation of fusion strategies: Results for UCID data set

12.3.2 Benchmarking-based Evaluation

The last missing building block to calculate the quality measures by the `trec_eval` tool are the trecval files for each data set. In order to assemble those files, specific queries have to be executed by each fusion strategy. In detail, one evaluation run consists of the selection of a random image from a topic. The generated result is then compared to the ground truth stored in the qrel file. For each topic, 50 runs have been applied to minimize noise.

Before diving into the results of the evaluations, the characteristics of the image collections will be discussed. From a structural point of view, both image collections are orthogonal to each other. The UCID collection exhibits over 200 topics on an overall amount of approx. 1300 images. This leads to a very sparse population of images in each topic. In contrast to that, 1000 images of the Wang collection are constantly and explicitly divided among ten topics. In this evaluation, the following quality measures have been calculated by `trec_eval`: Precision@20, recall@20, f-measure, and nDCG. Box plots illustrate the results of the evaluation runs as follows.

Figure 12.9 illustrates four box plots for the UCID evaluation. It is observable that Round Robin produces very low values for each quality measures. This is due to the fact, that it merges the results by simply using the rankings of the images in each result set. In terms of precision@20, the Late Fuzzy Multimedia Fusion as

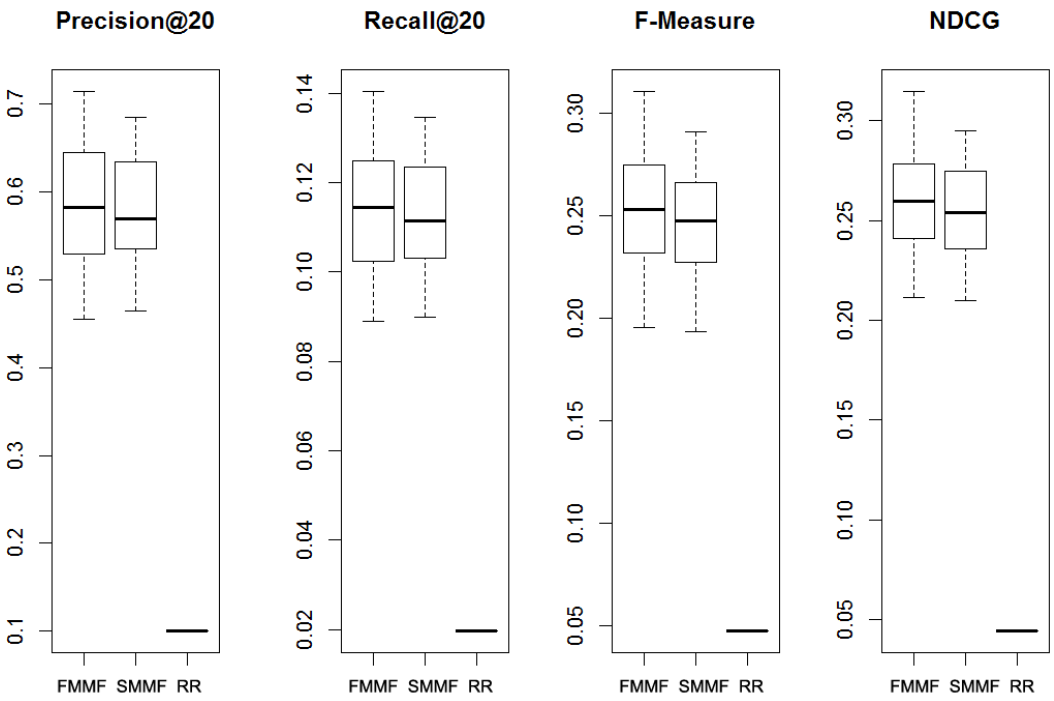


Figure 12.10: Evaluation of fusion strategies: Results for Wang data set

well as the simple multimedia fusion seems to produce low values as well in the first place, too. Due to the sparse population of images in a single topic, only a maximal reachable value of 0.50 for a topic containing 10 images or 0.25 for a topic covering 5 images in the ideal case looking at the 20 top ranked images. This softens the observation for precision@20. In contrast to that, recall@20 for both is acceptable. Obviously, the calculated f-measure is highly affected by the low precision@20 values. Besides, the nDCG is also affected by the situation of the topic to image ratio. Nevertheless, in all cases the Late Fuzzy Multimedia Fusion technique outperforms the simple multimedia fusion approach.

Figure 12.10 shows the four box plots for the Wang data set. Here, Round Robin adds no reasonable value for the evaluation by reaching very low values for all measures due to missing content-dependent reranking information. In contrast to the UCID evaluation, the precision@20 is for content-aware fusion strategies in a reasonable shape. Both reach values around 60% as average and up to 70% in the upper quartile. In this observation, the top twenty images cannot retrieve all relevant images resulting in an ideal recall@20 value of 0.20. The f-measure is lowered accordingly. In this case, the nDCG is more stable for both fusion algorithms. Following the observations of the UCID evaluation, the Late Fuzzy Multimedia Fusion performs slightly better for all quality measures compared to the simple multimedia fusion approach as well.

12.3.3 User-centric Evaluation

To have a more concrete view on the proposed approach with a random generated data set, a qualitative comparison of Round Robin, simple multimedia fusion and Late Fuzzy Multimedia Fusion has been conducted in a user study. 50 non-expert users have estimated the top 15 elements of the produced result sets by adding relevance scores res_i to the images: $res_i \in \{0, 1, 2, 3, 4, 5\}$, with 5 most relevant and 0 irrelevant. Based on this, DCG/nDCG have been calculated. The results are shown in Figure 12.11.

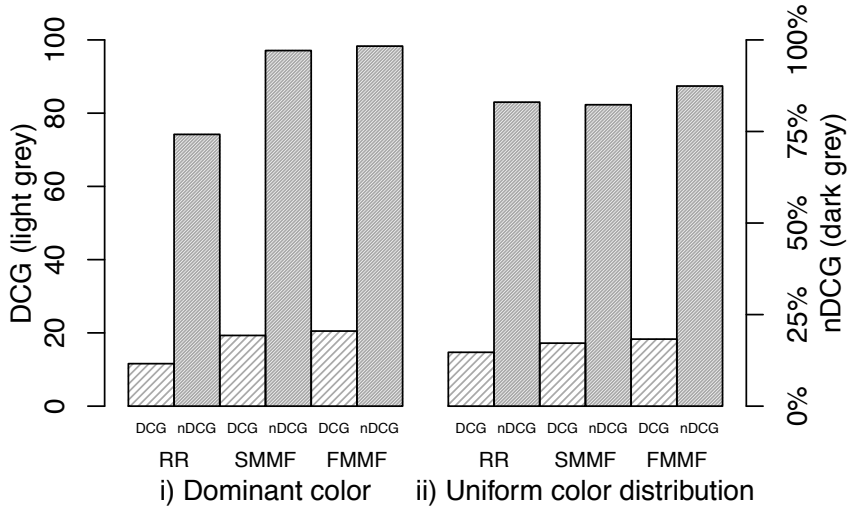


Figure 12.11: Qualitative evaluation of fusion strategies

In this regard, the IDCG for the top-15 images is 33. It has to be stated that one third of this value is produced by the relevance scores of the three top placed images. This is due to the logarithmic reduction factor of the DCG, which minimizes the impact of relevance scores of images with higher positions. The results for query a show that Round Robin is outperformed by the context-aware fusion strategies, because DCG and nDCG values are low. Late Fuzzy Multimedia Fusion performs slightly better than simple multimedia fusion, because the neighborhood distances have no high impact in image sets with a dominant colour. In contrast to that, the results of query b exhibit that Late Fuzzy Multimedia Fusion produces considerable better results than simple multimedia fusion. Here images with a uniform colour distribution are present that bootstrap the neighbourhood distance. The slight difference of 6% between nDCG values of Late Fuzzy Multimedia Fusion and simple multimedia fusion is caused by the logarithmic reduction factor. The algorithm produces only two rerankings in the top 5 images but lots of rerankings in the remaining results. The rerankings from position 6 to 15 improve the overall result set significantly, but are only minorly recognized by nDCG. An excerpt of the user evaluation showing the top-5 results for each fusion strategy can be found in Appendix C.3.

To complete this section, a performance evaluation of the fusion strategies is shown in Figure 12.12. Here, various sizes of the result sets show the ascent of the execution time.

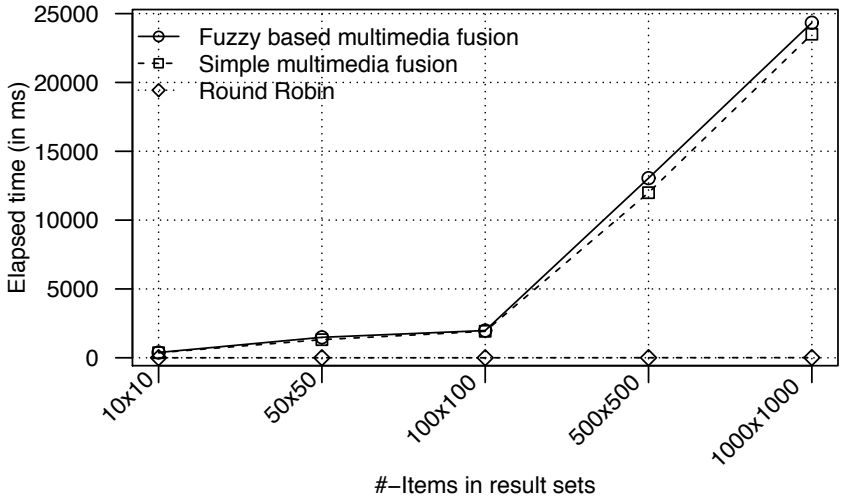


Figure 12.12: Performance evaluation of fusion strategies

Without doubt, Round Robin performs best with a (nearly) constant speed. Both, simple multimedia fusion and Late Fuzzy Multimedia Fusion have from result set size 100x100 a linear growth, with Late Fuzzy Multimedia Fusion taking a slight overhead (approximately 100 ms/150 ms for 500x500/1000x1000). In content centric strategies, the feature extraction is the bottleneck of processing leading to higher processing times.

Part VI

Summary

Chapter 13

Résumé

13.1 Conclusion

This thesis focused on the development of various techniques and prototypes to improve the current situation in the topic of unified and interoperable multimedia retrieval in distributed environments. In this regard, the three contributions will be shortly recapitulated and summarized: I have substantially contributed to a community-driven standardization process creating a multimedia annotation model and API to improve *multimedia metadata interoperability*. Here, I was involved as main editor of both W3C recommendations as well as published (journal) articles. Besides community activities, I developed the AIR framework from a conceptional point of view building the basement for *unified multimedia retrieval*. To ensure efficiency and an overall quality of the retrieval, I have integrated *optimization techniques for multimedia retrieval* that enabled AIR to be operated in a broad context of application domains.

The evaluation in the prior chapter clearly shows that the presented optimization techniques for federated multimedia retrieval enables the AIR framework for efficient and sophisticated multimedia retrieval in a broad domain of application scenarios. Due to the adaptation of query processing paradigms to multimedia retrieval efficiency can be guaranteed. In detail, the retrieval process is equipped with a data and demand driven query execution process along with a rule-based query-planning component as pre-processing. In addition, the proposed semantic multimedia caching system lifts AIR to a full-fledged federated external metasearch engine leading to a significant speed-up in terms of the identified phases of distributed retrieval. As a consequence, AIR covers all requirements of the THESEUS: MEDICO use case and serves as a mediator between the participating retrieval services enabling a efficient, harmonized and unified retrieval in the *federated medical retrieval* application scenario.

The take away message of the proposed multimedia result fusion approach evaluation is that a sophisticated fusion process in isolated environments needs content-dependent analysis to ensure an acceptable overall retrieval quality. The evaluation clearly showed that the information encapsulated in the neighborhood distances heavily improved the overall fusion process. The observed drawback of the execution times in both content-dependent approaches is smoothened due to the fact that 75% of users are only interested in the fractional amount of result items presented in the first/top page of the results [JS06]. Moreover, most services restrict results to a certain amount of resources. In sum, the proposed result fusion approach enables

AIR to be operated in the *isolated image retrieval* application scenario as well.

13.2 Future Work

This section covers the future work for each of the three central contributions of this thesis. The remaining is subdivided in three parts accordingly:

Community activities related to interoperable multimedia annotations. As already stated in Chapter 6 both specifications went through the complete W3C standardization process and are now in a mature state. Currently an industrial uptake is envisioned and first contacts, e.g., Wells Fargo[112], are already established. In terms of the ontology and the API itself, the following tasks will be initiated: A mechanism will be investigated to extend the model to describe the media production as well as the content analysis processing chains. Here, a focus should lie on the consolidation of the annotations produces, especially with respect to low-level features and feature extraction chains. Further, user interactions foster feedback loops to validate the annotation quality; in addition a transparent result combination of enrichment processes will also include provenance chains serving as basis for user acceptance and trustworthiness.

Improvements of the AIR Framework and optimizations for multimedia retrieval. The current version of the AIR Framework can be seen as a prototypical implementation of MPQF. To fulfill the application scenarios defined in Chapter 1, only a subset of the industrial standard was needed. Upcoming versions of the AIR Framework will consider the integration of further MPQF features. Besides enlarging the MPQF support, the AIR framework will be extended by a hierarchical error message system, which is especially needed in distributed environments to have a more fine granular view on arising issues while evaluating federated queries. In terms of the query optimization the rules for query execution planning in future will apply more structural changes of a query and the integration of a more advanced cost model is planned. Future work will also focus on update mechanisms to keep the data in the query cache up to date as well as the integration of techniques to improve cold start issues of the query cache.

Extension of the Late Fuzzy Multimedia Fusion approach. The current version of the Late Fuzzy Multimedia Fusion approach offers great potential for further refinements. Future work will be on the one hand the integration of a preprocessing steps undertaking tasks like conducting a semantically clustering or further data mining concepts to find common representatives for clusters computable in very large data portions. On the other hand, the approach shall be enlarged to handle multimodal feature combinations in order to be more domain independent and follow the overall idea of the AIR framework to enable multi-modal retrieval.

[112] https://www.wellsfargo.com/, last checked December 18, 2013.

Another important aspect is to include learning algorithms as well as user feedback to establish a semi-automatic configuration of weighting factors as well as processing threads.

Appendix A

Fuzzy Logic

Fuzzy logic is a substitute of many-valued logic and has been firstly issued by Zadeh in 1965 [Zad65]. This section introduce only a subset of fuzzy logic and covers topics, which are substantial for this thesis. It is based on the book of Klir and Yuan [KY95]. To get an comprehensive overview over fuzzy logic as well as its mathematical foundations, the interested reader is guided to [KY95], [Cox93] or [GNW95].

Fuzzy logic can be seen as an extension of the Boolean logic. The binary membership evaluation of the classical Boolean logic is often termed as *crisp set*.

Definition 15 (*Crisp set*)

The crisp set is defined in such way as to dichotomize the individuals in some given universe of discourse into two groups: members and nonmembers. A sharp, unambiguous distinction exists between the members and nonmembers of the set.

Fuzzy logic enlarges the sharp characteristics of crisp sets in order to express uncertainty by a gradual transition from membership to nonmembership. Most commonly, the unit interval $[0, 1]$ is used to define the grade of the membership, with 1 donating full and 0 no membership.

Definition 16 (*Fuzzy set*)

Fuzzy set A in the universal set X is completely and uniquely defined by one particular membership function μ_A as follows:

$$\mu_A : X \to [0, 1]$$

The membership function μ_A assigns to every element of the universal set X a membership grade. In the domain of fuzzy sets, the three classical set operations of crisp sets are defined as follows:

Definition 17 (*Standard complement*)

The standard complement $\bar{A}$ of a fuzzy set A with respect to the universal set X is defined for all $x \in X$ as follows:

$$\bar{A}(x) = 1 - A(x)$$

Definition 18 (*Standard intersection*)

Given two fuzzy sets, A and B, their standard intersection $A \cap B$ are defined

for all $x \in X$ as follows:

$$(A \cap B)(x) = min\,[A(x), B(x)]$$

Definition 19 (*Standard union*)

Given two fuzzy sets, A and B, their standard intersection $A \cup B$ are defined for all $x \in X$ as follows:

$$(A \cup B)(x) = max\,[A(x), B(x)]$$

The introduced definitions of the classical operations are the *standard fuzzy set operations*. In literature, a variety of functions exist, that qualify as fuzzy generalizations of the classical operations. Beside fuzzy complement, specific classes for functions suitable for fuzzy intersection and fuzzy unions are termed *t-norms* and *t-conorms* in literature. These are formally defined as follows:

Definition 20 (*Fuzzy complement*)

A fuzzy complement c of a fuzzy set A is specified in general by an unary operation on the unit interval:

$$c : [0,1] \to [0,1].$$
$$c(A(x)) = cA(x),$$

for all $x \in X$, with X the universal set, $A(x)$ membership grade, $cA(x)$ the value of the complement. It satisfies at least the following axioms for all $a, b \in [0,1]$:

Axiom c1.	$c(0) = 1$ *and* $c(1) = 0$	*(boundary condition)*
Axiom c2.	$\forall a, b \in [0,1]$ *, if* $a \leq b$*, then* $c(a) \geq c(b)$	*(monotonicity)*

Definition 21 (*Fuzzy intersections / t-norm*)

A fuzzy intersection / t-norm i of two fuzzy sets A and B is specified in general by a binary operation on the unit interval:

$$i : [0,1] \times [0,1] \to [0,1].$$
$$(A \cap B)(x) = i\,[A(x), B(x)],$$

for all $x \in X$, with X the universal set and $A(x)$, $B(x)$ membership grades. It satisfies at least the following axioms for all $a, b, d \in [0,1]$:

Axiom i1.	$i(a,1) = a$	*(boundary condition)*
Axiom i2.	$b \leq d$ *implies* $i(a,b) \leq i(a,d)$	*(monotonicity)*
Axiom i3.	$i(a,b) = i(b,a)$	*(commutativity)*
Axiom i4.	$i\big(a, i(b,d)\big) = i\big(i(a,b), d\big)$	*(associativity)*

Definition 22 (*Fuzzy union / t-conorm*)

A fuzzy union / t-conorm u of two fuzzy sets A and B is specified in general by a binary operation on the unit interval:

$$u : [0,1] \times [0,1] \to [0,1].$$
$$(A \cup B)(x) = u\,[A(x), B(x)],$$

for all $x \in X$, with X the universal set and $A(x)$, $B(x)$ membership grades. It satisfies at least the following axioms for all $a, b, d \in [0,1]$:

Axiom u1.	$u(a,1) = a$	*(boundary condition)*
Axiom u2.	$b \leq d$ *implies* $u(a,b) \leq u(a,d)$	*(monotonicity)*
Axiom u3.	$u(a,b) = u(b,a)$	*(commutativity)*
Axiom u4.	$u\big(a,u(b,d)\big) = u\big(u(a,b),d\big)$	*(associativity)*

For fuzzy complement, t-norm as well as t-conorm the *axiomatic skeleton* has been defined. There exist also additional restrictions, but a further consideration is not in the scope of this thesis. It has to be stated, that the standard fuzzy intersection and union are the only idempotent t-norms and t-conorms.

Appendix B

Details on Ontology & API for Media Resource 1.0

B.1 Ontology for Media Resource 1.0: Properties

This appendix highlights the properties of the Ontology for Media Resource 1.0. Along with basic information, their subtypes are introduced and the specific semantics will be defined. The values in Tables B.1 to B.8 are specified by primitive datatypes of XML Schema definition [BM04]. In the descriptions, "|" donates *or* and key_{opt} an optional property.

Table B.1: Media identification properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**
identifier	key="identifier", value=anyURI;	A URI identifying a media resource.
title	key="title", value=string; key_{opt}="type", value=(anyURI | string);	A tuple specifying the title of a media resource as plain text and an optional type parameter of the title, e.g., subtitle as URI or in plain text.
language	key="language", value=(anyURI | string);	The language used in the media resource. Recommended best practice is [Alv95] as controlled vocabulary.
locator	key="locator", value=anyURI;	A reachable URI to access the media resource.

Table B.2: Creation descriptive properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**
contributor	key="contributor", value=(anyURI | string); key_{opt}="role", value=(anyURI | string);	A person can be specified by either URI (recommended best practice) or a plain text. Optionally, the specific role can be defined by either URI (recommended best practice) or a plain text.
creator	key="creator", value=(anyURI | string); key_{opt}="role", value=(anyURI | string);	A person can be specified by either URI (recommended best practice) or a plain text. Optionally, the specific role can be defined by either URI (recommended best practice) or a plain text.

date	key="date", value=date;
	key_{opt}="type",
	value=(anyURI | string);
		A date specifies a timestamp related to a media resource.
		Optionally, the specific date can be defined by either URI (recommended best practice) or a plain text.

location	key="name",
	value=(anyURI | string);
	key_{opt}="longitude",
	value=decimal;
	key_{opt}="latitude",
	value=decimal;
	key_{opt}="altitude",
	value=decimal;
	key_{opt}="coordinate-System",
	value=(anyURI | string);
		A location can be specified either by a URI (recommended best practice) or a plain text. It describes where the resource has been created or assembled.
		Optionally, a complete geographic positioning can be applied by longitude, latitude and altitude. The coordinate system is use can be applied either by a URI (recommended best practice) or a plain text.

Table B.3: Content descriptive properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**
description	key="description", value=string;	Plain text summarizing the content of the media resource.
keyword	key="keyword", value=(anyURI | string);	Concepts, that describe the media resource best. A keyword can be either a URI (recommended best practice) or a plain text.
genre	key="genre", value=(anyURI | string);	The categories of the media resource are defined by either a URI (recommended best practice) or a plain text.

rating	key="value",
	value=Decimal;
	key="ratingSystem",
	value=(anyURI | string);
	key_{opt}="min",
	value=decimal;
	key_{opt}="max",
	value=decimal;
		The rating of a media resource is defined by a decimal value and the used rating system (e.g., 5-star-rating). The rating system can be either a URI (recommended best practice) or a plain text.
		Optionally, the minimum as well as maximum rating value can be specified.

Table B.4: Relational properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**

relation	key="target",
	value=(anyURI | String);
	key$_{opt}$="type",
	value=(anyURI | string);
		A media resource can be defined as related to the current one by either a URI (recommended best practice) or a plain text.
		Optionally, the type of relation can be specified by either a URI (recommended best practice) or a plain text.

collection	key="collection", value=(anyURI | string);	The name of the collection the media resource stems from defined by either a URI (recommended best practice) or a plain text.

Table B.5: Rights properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**

copyright	key="copyright",
	value=string;
	key$_{opt}$="holder",
	value=(anyURI | string);
		The copyright statement that is associated with the media resource.
		Optionally, the copyright holder can be defined by either a URI (recommended best practice) or a plain text.

policy	key="statement", value=(anyURI | string); key$_{opt}$="type", value=(anyURI | string);	The policy statement that is associated with the media resource. Optionally, the type can be defined by either a URI (recommended best practice) or a plain text.

Table B.6: Distribution properties of the Ontology for Media Resource 1.0

Metadata property	Type definition	Description
publisher	key="publisher", value=(anyURI | string);	The publisher of the media resource defined by either a URI (recommended best practice) or a plain text.
targetAudience	key="audience", value=(anyURI | string); key_{opt}="classificationSystem", value=(anyURI | string);	Specifies a audience for which the media resource has been produced. Optionally, a classification system can be specified by either a URI (recommended best practice) or a plain text.

Table B.7: Fragmentation properties of the Ontology for Media Resource 1.0

Metadata property	Type definition	Description
fragment	key="identifier", value=anyURI; key_{opt}="role", value=(anyURI | string);	A URI-based identifier for a specific portion of a media resource. Recommended best practice is [TMPD12]. Optionally, the role of the fragment can be specified by either a URI (recommended best practice) or a plain text.
namedFragment	key="identifier", value=anyURI; key="label", value=string;	A URI-based identifier for a specific portion of a media resource. Recommended best practice is [TMPD12]. The label is a human readable name of the media resource.

Table B.8: Technical properties of the Ontology for Media Resource 1.0

Metadata property	**Type definition**	**Description**
frameSize	key="width", value=decimal; key"height", value=decimal; key_{opt}="unit", value=string;	The frame size is specified by values for width and height. Optionally, a unit can be defined, if not, it must be interpreted as pixels.
compression	key="compression", value=(anyURI | string);	The compression type used in the media resource.
duration	key="duration", value=decimal;	The duration of the media resource in seconds.
format	key="format", value=(anyURI | string);	The format of the media resource can be specified by either a URI (recommended best practice) or a plain text.
samplingRate	key="samplingRate", value=decimal;	The sampling rate of the media resource is defined as decimal.
frameRate	key="frameRate", value=decimal;	The frame rate of the media resource is defined as decimal.
averageBitRate	key="averageBitRate", value=decimal;	The average bit rate of the media resource is defined as decimal.
numTracks	key="number", value=double; key="type", value=string;	The number of tracks of the media resource is defined as double.

B.2 API for Media Resource 1.0: WebIDL specification

Listing B.1 illustrates the complete API for Media Resource 1.0 in WebIDL.

```
interface MediaResource {
  short getSupportedModes();
  MediaResource createMediaResource(
      DOMString mediaResource,
      optional MetadataSource[] metadataSources,
      optional short mode
  );
};

interface AsyncMediaResource : MediaResource {
  void getMediaProperty(
      DOMString[] propertyNames,
      PropertyCallback successCallback,
      ErrorCallback errorCallback,
      optional DOMString fragment,
      optional DOMString sourceFormat,
      optional DOMString language
  );
  void getOriginalMetadata(
      DOMString sourceFormat,
      MetadataCallback successCallback,
      ErrorCallback errorCallback
  );
};

interface PropertyCallback {
    void handleEvent (
      MediaAnnotation[] mediaAnnotations
    );
};

interface MetadataCallback {
   void handleEvent (
      DOMString[] metadata
   );
};

interface ErrorCallback {
   void handleEvent (DOMString errorStatus);
};

interface SyncMediaResource : MediaResource {
    MediaAnnotation[] getMediaProperty(
      DOMString[] propertyNames,
      optional DOMString fragment,
      optional DOMString sourceFormat,
      optional DOMString language
    );
    DOMString[] getOriginalMetadata(
      DOMString sourceFormat
```

```
51      );
52  };
53
54  interface MetadataSource {
55     attribute DOMString metadataSource;
56     attribute DOMString sourceFormat;
57  };
58
59  interface MediaAnnotation {
60     attribute DOMString propertyName;
61     attribute DOMString value;
62     attribute DOMString language;
63     attribute DOMString sourceFormat;
64     attribute DOMString fragmentIdentifier;
65     attribute DOMString mappingType;
66     attribute short     statusCode;
67  };
68
69  interface Identifier : MediaAnnotation {
70     attribute DOMString identifierLink;
71  };
72
73  interface Title : MediaAnnotation {
74     attribute DOMString titleLabel;
75     attribute DOMString typeLink;
76     attribute DOMString typeLabel;
77  };
78
79  interface Language : MediaAnnotation {
80     attribute DOMString languageLink;
81     attribute DOMString languageLabel;
82  };
83
84  interface Locator : MediaAnnotation {
85     attribute DOMString locatorLink;
86  };
87
88  interface Contributor : MediaAnnotation {
89     attribute DOMString contributorLink;
90     attribute DOMString contributorLabel;
91     attribute DOMString roleLink;
92     attribute DOMString roleLabel;
93  };
94
95  interface Creator : MediaAnnotation {
96     attribute DOMString creatorLink;
97     attribute DOMString creatorLabel;
98     attribute DOMString roleLink;
99     attribute DOMString roleLabel;
100 };
101
102 interface MADate : MediaAnnotation {
103    attribute DOMString date;
104    attribute DOMString typeLink;
```

```
105     attribute DOMString typeLabel;
106 };
107
108 interface Location : MediaAnnotation {
109     attribute DOMString locationLink;
110     attribute DOMString locationLabel;
111     attribute double    longitude;
112     attribute double    latitude;
113     attribute double    altitude;
114     attribute DOMString coordinateSystemLabel;
115     attribute DOMString coordinateSystemLink;
116 };
117
118 interface Description : MediaAnnotation {
119     attribute DOMString descriptionLabel;
120 };
121
122 interface Keyword : MediaAnnotation {
123     attribute DOMString keywordLink;
124     attribute DOMString keywordLabel;
125 };
126
127 interface Genre : MediaAnnotation {
128     attribute DOMString genreLink;
129     attribute DOMString genreLabel;
130 };
131
132 interface Rating : MediaAnnotation {
133     attribute double    ratingValue;
134     attribute DOMString ratingSystemLink;
135     attribute DOMString ratingSystemLabel;
136     attribute double    min;
137     attribute double    max;
138 };
139
140 interface Relation : MediaAnnotation {
141     attribute DOMString targetLink;
142     attribute DOMString targetLabel;
143     attribute DOMString typeLink;
144     attribute DOMString typeLabel;
145 };
146
147 interface Collection : MediaAnnotation {
148      attribute DOMString collectionLink;
149      attribute DOMString collectionLabel;
150 };
151
152 interface Copyright : MediaAnnotation {
153     attribute DOMString copyrightLabel;
154     attribute DOMString holderLink;
155     attribute DOMString holderLabel;
156 };
157
158 interface Policy : MediaAnnotation {
```

```
159     attribute DOMString statementLink;
160     attribute DOMString statementLabel;
161     attribute DOMString typeLink;
162     attribute DOMString typeLabel;
163  };
164
165  interface Publisher : MediaAnnotation {
166     attribute DOMString publisherLink;
167     attribute DOMString publisherLabel;
168  };
169
170  interface TargetAudience : MediaAnnotation {
171     attribute DOMString audienceLink;
172     attribute DOMString audienceLabel;
173     attribute DOMString classificationSystemLink;
174     attribute DOMString classificationSystemLabel;
175  };
176
177  interface Fragment : MediaAnnotation {
178     attribute DOMString identifier;
179     attribute DOMString roleLink;
180     attribute DOMString roleLabel;
181  };
182
183  interface NamedFragment : MediaAnnotation {
184     attribute DOMString identifier;
185     attribute DOMString label;
186  };
187
188  interface FrameSize : MediaAnnotation {
189     attribute double    width;
190     attribute double    height;
191     attribute DOMString unit;
192  };
193
194  interface Compression : MediaAnnotation {
195      attribute DOMString compressionLink;
196      attribute DOMString compressionLabel;
197  };
198
199  interface Duration : MediaAnnotation {
200      attribute double duration;
201  };
202
203  interface Format : MediaAnnotation {
204      attribute DOMString formatLink;
205      attribute DOMString formatLabel;
206  };
207
208  interface SamplingRate : MediaAnnotation {
209      attribute double samplingRate;
210  };
211
212  interface FrameRate : MediaAnnotation {
```

```
213     attribute double frameRate;
214 };
215
216 interface AverageBitRate : MediaAnnotation {
217     attribute double averageBitRate;
218 };
219
220 interface NumTracks : MediaAnnotation {
221      attribute short      number;
222      attribute DOMString typeString;
223 };
```

Listing B.1: WebIDL specification of API for Media Resource 1.0

Appendix C

Details on the Evaluation of the AIR Framework

C.1 Query Cache Evaluation: Query visualizations & Processing times

This section introduces the queries used in the benchmarking suite of the query cache. Figure C.1 illustrates the query structure as well as their correlation to each other. Table C.1 summarizes the execution times of the benchmarking suite with varying configuration parameter of the benchmark suite.

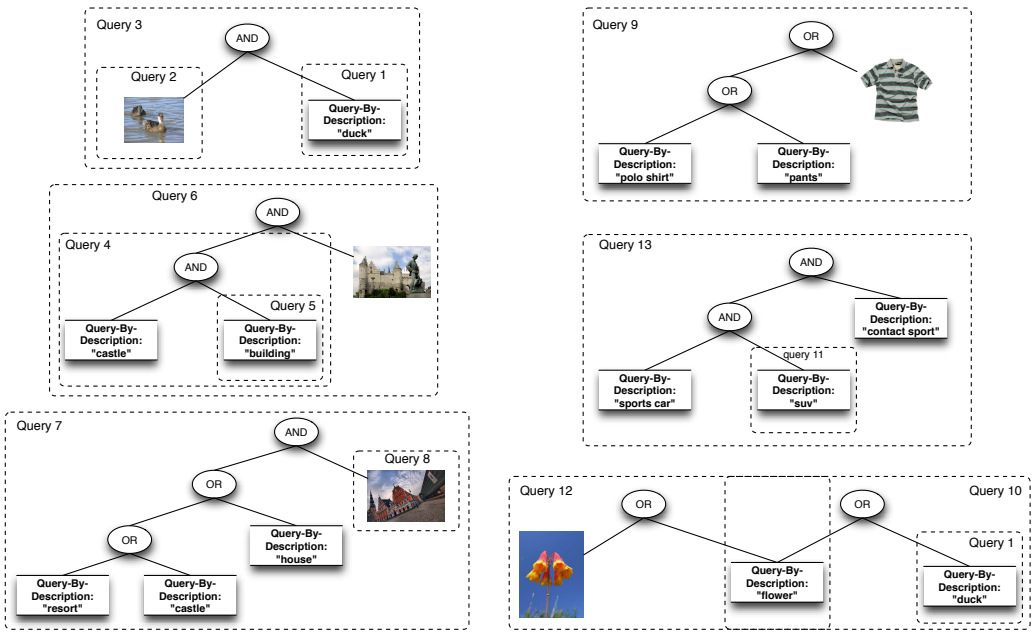


Figure C.1: Queries and their correlation used for evaluation of multimedia caching system

Table C.1: Multimedia caching system: median query duration in seconds for complete benchmark suite

Limiter/Interval	100	150	200	250
1 - no cache	1432.48	1254.95	884.47	646.02
2 - no cache	485.61	248.41	5.31	3.83
4 - no cache	161.43	4.00	3.77	3.76
8 - no cache	54.85	4.05	3.79	3.77
16 - no cache	168.73	4.92	3.91	3.78
1 - active cache	66.93	1.48	0.90	0.77
2 - active cache	4.57	0.87	0.76	0.76
4 - active cache	3.82	0.82	0.74	0.73
8 - active cache	4.99	0.83	0.75	0.72
16 - active cache	2.90	0.84	0.78	0.71

C.2 Query Processing Strategies: Detailed Processing Times

This section includes detailed boxplots (Figure C.2 to C.6) for the five queries specified in Section 12.2.1 of the query processing strategies. In all boxplots, the y-axis denotes the runtime in milliseconds that was needed to evaluate a query. The runtime is the average of 25 executions for each measurement. The x-axis denotes the number of result items in every leaf of the query tree.

C.3 Late Fuzzy Multimedia Fusion: Excerpt of the User Evaluation

Figure C.7 (dominant color) and C.8 (uniform color distribution) show an excerpt of the performed user evaluation along with the corresponding query.

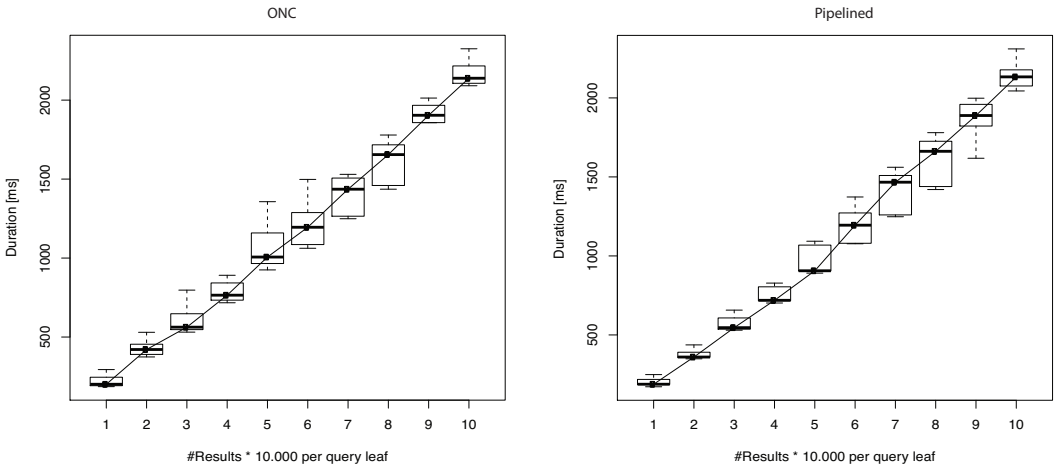


Figure C.2: Boxplot comparing execution times for demand- and data-driven query processing strategies: Query I

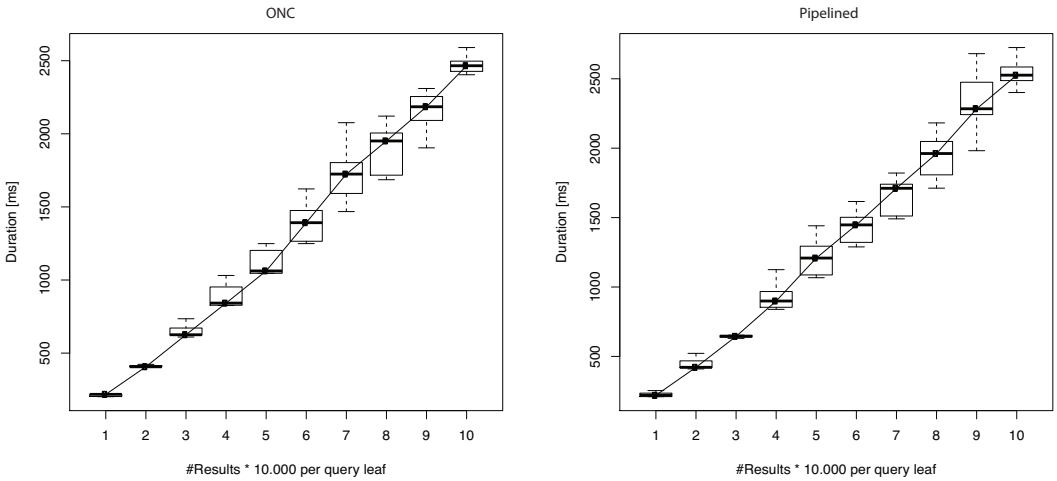


Figure C.3: Boxplot comparing execution times for demand- and data-driven query processing strategies: Query II

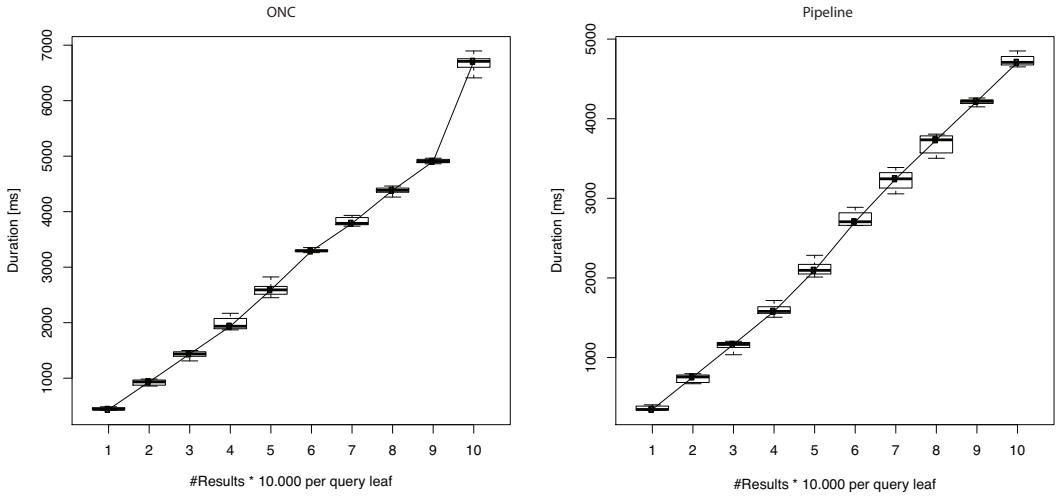


Figure C.4: Boxplot comparing execution times for demand- and data-driven query processing strategies: Query III

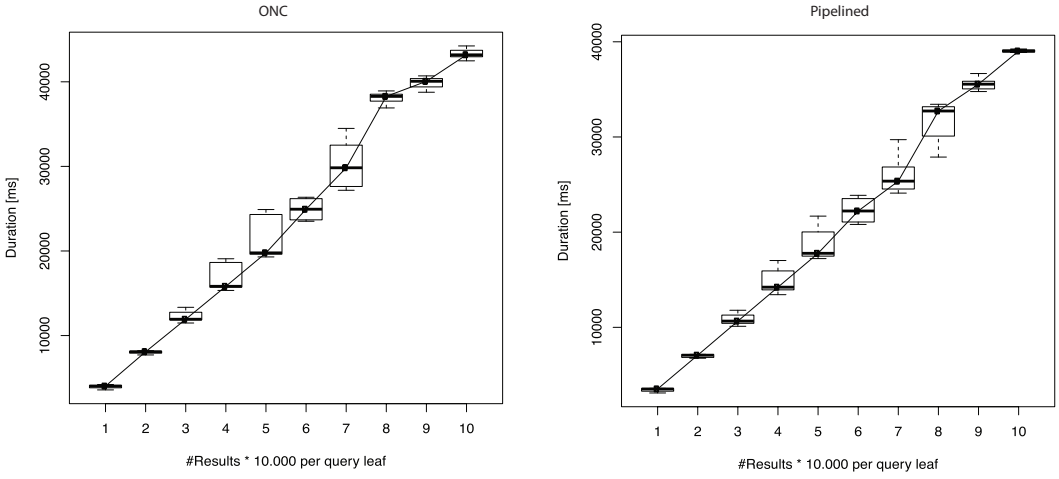


Figure C.5: Boxplot comparing execution times for demand- and data-driven query processing strategies: Query IV

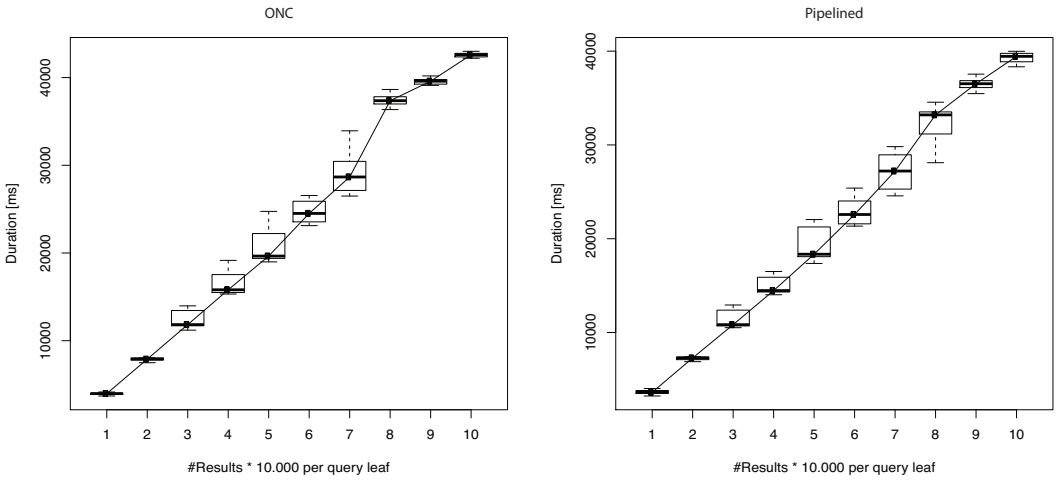


Figure C.6: Boxplot comparing execution times for demand- and data-driven query processing strategies: Query V

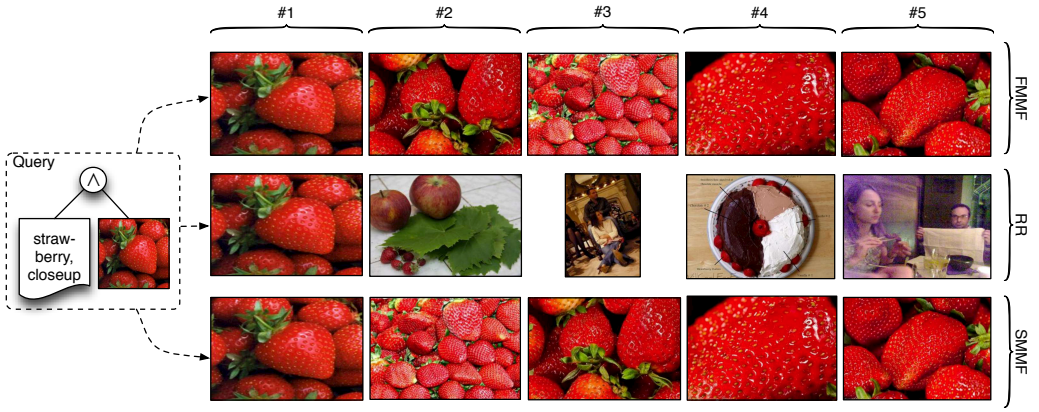


Figure C.7: Excerpt of the dominant color evaluation: Top-5 results of evaluated fusion strategies

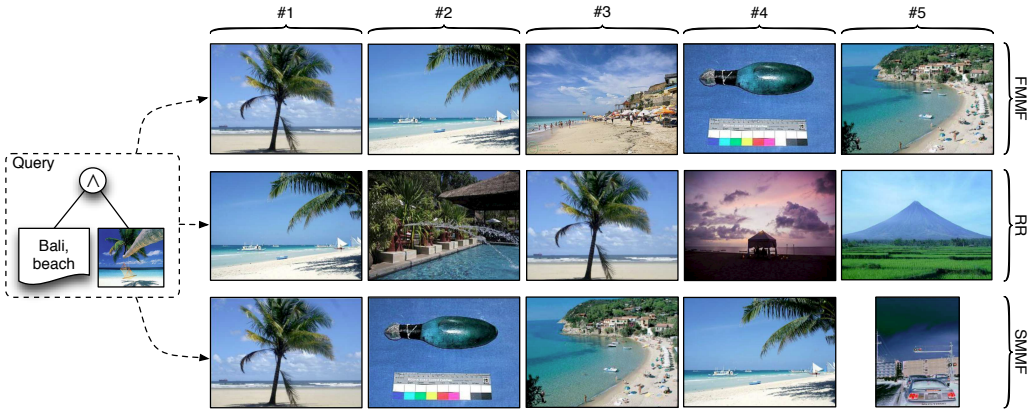


Figure C.8: Excerpt of the uniform color distribution evaluation: Top-5 results of evaluated fusion strategies

Bibliography

[ABK01] Solomon Atnafu, Lionel Brunie, and Harald Kosch. Similarity-based algebra for multimedia database systems. In *Proceedings of the 12th Australasian Database Conference*, pages 115–122, 2001. (Cited on page 53.)

[ABMD92] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205 –220, 1992. (Cited on page 40.)

[ACB02] Solomon Atnafu, Richard Chbeir, and Lionel Brunie. Efficient content-based and metadata retrieval in image database. *Journal of Universal Computer Science*, 8(6):613–622, June 2002. (Cited on page 17.)

[AFS93] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. *Foundations of Data Organization and Algorithms*, 730:69–84, 1993. (Cited on page 42.)

[AH08] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. The MIT Press, Cambridge Massachusetts, 2008. (Cited on page 30.)

[AHSB12] Ben Adida, Ivan Herman, Manu Sporny, and Mark Birbeck. RDFa 1.1 Primer - Rich Structured Data Markup for Web Documents. W3C Recommendation. 07 June, 2012. `http://www.w3.org/TR/xhtml-rdfa-primer/`. (Cited on page 33.)

[All07] John Allsopp. *(M)icroformats: Empowering Your Markup for Web 2.0*. Friends of Ed, Berkeley, CA, 2007. (Cited on page 33.)

[Alv95] H. Alvestrand. Tags for the identification of languages. RFC 1766 (Proposed Standard), March 1995. Obsoleted by RFCs 3066, 3282. (Cited on page 145.)

[AMR+12] Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, and Pierre Senellart. *Web Data Management*. Cambrigde Universit Press, New York, NY, USA, 2012. (Cited on page 27.)

[And10] Peter Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Springer, 2nd edition, 2010. (Cited on page 30.)

[ANS95] ANSI/NISO Z39.50-1995. Information retrieval (Z39.50): Application service definition and protocol specification. July, 1995. `http://www.kbr.be/bezig/part1.pdf`. (Cited on page 15.)

[ANS05] ANSI/NISO Z39.19-2005. Guidelines for the construction, format, and management of monolingual controlled vocabularies. July, 2005. http://www.niso.org/standards/z39-19-2005/. (Cited on page 22.)

[APC05] James D. Anderson and José Pérez-Carballo. *Information retrieval design: Principles and options for information description, organization, display, and access in information retrieval databases, digital libraries, catalogs and indexes*. University Publishing Solutions, LLC, East Brunswick, NJ, USA, 2005. (Cited on page 12.)

[AVL62] G. Adelson-Velskii and E. M. Landis. An algorithm for the organization of information. *Proceedings of the USSR Academy of Sciences*, 146(2):263–266, 1962. (Cited on page 45.)

[Bay72] Rudolf Bayer. Symmetric binary B-Trees: Data structure and maintenance algorithms. *Acta Informatica*, 1:290–306, 1972. (Cited on page 44.)

[BBD+08] Werner Bailer, Lionel Brunie, Mario Döller, Michael Granitzer, Ralf Klamma, Harald Kosch, Mathias Lux, and Marc Spaniol. Multimedia metadata standards. In Borko Furht, editor, *Encyclopedia of Multimedia*, pages 568–575. Springer US, 2008. (Cited on pages 21 and 22.)

[BBK98] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegl. The pyramid-technique: towards breaking the curse of dimensionality. In *Proceedings of the ACM International Conference on Management of Data*, pages 142–153, New York, NY, USA, 1998. ACM. (Cited on page 47.)

[BBK01] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Survey*, 33(3):322–373, 2001. (Cited on page 45.)

[BBP03] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Merging results of distributed image libraries. In *Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 33–36, Baltimore, Maryland, USA, 2003. IEEE Computer Society. (Cited on page 127.)

[Bel05] John L. Bell. *Set Theory: Boolean-Valued Models and Independence Proofs*. Oxford University Press, 3^{rd} edition, 2005. (Cited on page 14.)

[BG04] Dan Brickley and Ramanathan Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. 10 February, 2004. http://www.w3.org/TR/rdf-schema/. (Cited on page 30.)

[BGBR+10] Stanislav Barton, Valérie Gouet-Brunet, Marta Rukoz, Christophe Charbuillet, and Geoffroy Peeters. Estimating the indexability of multimedia descriptors for similarity searching. In *Proceedings of International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 84–87. CID, 2010. (Cited on page 47.)

[BHKM05] Matthias Brantner, Sven Helmer, Carl-Christian Kanne, and Guido Moerkotte. Full-fledged algebraic XPath processing in Natix. In *Proceedings of the 21st International Conference on Data Engineering*, pages 705–716, 2005. (Cited on page 53.)

[BHL+09] Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, and Henry Thompson. Namespaces in XML 1.0 (Third Edition). W3C Recommendation. 08 December, 2009. http://www.w3.org/TR/REC-xml-names/. (Cited on page 28.)

[Bim99] Alberto Del Bimbo. *Visual information retrieval*. Morgan Kaufmann, 1999. (Cited on page 24.)

[BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-Tree: An efficient and robust access method for points and rectangles. In *Proceedings of the International Conference on Management of Data*, pages 322–331. ACM Press, 1990. (Cited on page 46.)

[Bla11] Paul E. Black. Binary search tree. Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST), 2011. http://www.nist.gov/dads/HTML/binarySearchTree.html. (Cited on page 45.)

[BLFM98] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396 (Draft Standard), August 1998. Obsoleted by RFC 3986, updated by RFC 2732. (Cited on page 28.)

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284:34–43, 2001. (Cited on page 29.)

[BM04] Paul V. Biron and Ashok Malhotra. XML Schema part 2: Datatypes second edition. W3C Recommendation 28 October, 2004. http://www.w3.org/TR/xmlschema-2/s. (Cited on page 145.)

[BPSM+08] Tim Bray, Jean Paoli, Michael Sperberg-McQueen, Eve Maler, and Francois Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation. 26 November, 2008. http://www.w3.org/TR/2008/REC-xml-20081126/. (Cited on page 27.)

[Bry85] Victor Bryant. *Metric Spaces: Iteration and Application*. Cambridge University Press, Melbourne, Australia, 4^th edition, 1985. (Cited on page 42.)

[BS09] Norbert Beckmann and Bernhard Seeger. A revised R*-Tree in comparison with related index structures. In *Proceedings of the International Conference on Management of Data*, pages 799–812. ACM, 2009. (Cited on page 48.)

[BW80] J. L. Bentley and D. Wood. An optimal worst case algorithm for reporting intersections of rectangles. *IEEE Transactions on Computers*, 29(7):571–577, July 1980. (Cited on page 45.)

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*. Addison Wesley Longman Publishing Co. Inc., 1999. (Cited on page 13.)

[CCB09] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32 th International Conference on Research and Development in Information Retrieval*, pages 758–759, 2009. (Cited on page 111.)

[CCL08] Wei Chen, Jing Chen, and Qing Li. Adaptive community-based multimedia data retrieval in a distributed environment. In *Proceedings of the 2^{nd} Conference on Ubiquitous Information Management and Communication*, pages 20–24, 2008. (Cited on pages 78 and 95.)

[CG00] Jason P. A. Charlesworth and Philip N. Garner. Spoken content metadata and MPEG-7. In *Proceedings of the 8^{th} ACM International Conference on Multimedia*, pages 81–84, 2000. (Cited on page 18.)

[Cha98] Surajit Chaudhuri. An overview of query optimization in relational systems. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS'98)*, pages 34–43, 1998. (Cited on page 52.)

[Cha02] Donald D. Chamberlin. XQuery: An XML query language. *IBM Systems Journal*, 41(4):597–615, 2002. (Cited on page 53.)

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007. (Cited on page 42.)

[Chr85] Stavros Christodoulakis. Multimedia database management systems (panel). In *SIGMOD Conference*, pages 304–305, 1985. (Cited on page 49.)

[Cla02] David A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):45–62, 2002. (Cited on page 41.)

[CMP02] Donatella Castelli, Carlo Meghini, and Pasquale Pagano. Foundations of a multidimensional query language for digital libraries. volume 2458 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2002. (Cited on page 52.)

[Cod70] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13:377–387, June 1970. (Cited on pages 12 and 53.)

[Cox93] Earl Cox. *The fuzzy systems handbook: A practitioner's guide to building, using, and maintaining fuzzy systems*. Academic Press, 1993. (Cited on page 141.)

[CQL11] Xinlei Chen, Xinquan Qu, and Zijian Li. Image analysis with nonlinear adaptive dimension reduction. In *Proceedings of the International Conference on Internet Multimedia Computing and Service*, ACM International Conference Proceeding Series, pages 134–137. ACM, 2011. (Cited on page 47.)

[Cro06] D. Crockford. The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627 (Informational), July 2006. (Cited on page 29.)

[CS10] Kasim Selcuk Candan and Maria Luisa Sapino. *Data management for multimedia retrieval*. Cambridge University Press, New York, NY, USA, 2010. (Cited on pages 16 and 39.)

[CTZZ10] Bin Cui, Anthony K.H. Tung, Ce Zhang, and Zhe Zhao. Multiple feature fusion for social media applications. In *Proceedings of the International Conference on Management of Data*, pages 435–446. ACM, 2010. (Cited on page 48.)

[CXH04] Isabel Cruz, Huiyong Xiao, and Feihong Hsu. An ontology-based framework for XML semantic integration. In *Proceedings of the International Database Engineering and Applications Symposium*, pages 217–226, 2004. (Cited on page 59.)

[Dad96] Peter Dadam. *Verteilte Datenbanken und Client/Server-Systeme: Grundlagen, Konzepte und Realisierungsformen*. Springer-Verlag Berlin Heidelberg, 1996. (Cited on page 89.)

[DAE07] Frederic Dufaux, Michael Ansorge, and Touradj Ebrahimi. Overview of JPSearch: a standard for image search and retrieval. In *Proceedings of the 5^{th} International Workshop on Content-based Multimedia Indexing*, pages 138–143, Bordeaux, France, 2007. (Cited on page 59.)

[Dam64] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964. (Cited on page 106.)

[DBKG08] Mario Döller, Kerstin Bauer, Harald Kosch, and Matthias Gruhne. Standardized multimedia retrieval based on Web Service technologies and the MPEG Query Format. *Journal of Digital Information*, 6(4):315–331, 2008. (Cited on page 78.)

[DD11] Danny Dover and Erik Dafforn. *Search Engine Optimization (SEO) Secrets*. Wiley Publishing, 1st edition, 2011. (Cited on page 32.)

[DGB06] Chabane Djeraba, Moncef Gabbouj, and Patrick Bouthemy. Multimedia indexing and retrieval: Ever great challenges. *Multimedia Tools Applications*, 30(3):221–228, September 2006. (Cited on page 44.)

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(3):1–60, 2008. (Cited on page 6.)

[DKM09] Mario Döller, Harald Kosch, and Paul Maier. Image database. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1353–1358. Springer US, 2009. (Cited on page 17.)

[DKN08] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2):77–107, 2008. (Cited on page 42.)

[DLKS11] Mario Döller, Sebastian Lehrack, Harald Kosch, and Ingo Schmitt. Quantum logic based MPEG Query Format algebra. In *Adaptive Multimedia Retrieval. Context, Exploration, and Fusion*, volume 6817 of *Lecture Notes in Computer Science*, pages 204–219. 2011. (Cited on page 53.)

[DMR+12] Duo Ding, Florian Metze, Shourabh Rawat, Peter Franz Schulam, Susanne Burger, Ehsan Younessian, Lei Bao, Michael G. Christel, and Alexander G. Hauptmann. Beyond audio and video retrieval: Towards multimedia summarization. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pages 2:1–2:8. ACM, 2012. (Cited on page 48.)

[DSK+10] Mario Döller, Florian Stegmaier, Harald Kosch, Ruben Tous, and Jaime Delgado. Standardized interoperable image retrieval. In *Proceedings of the ACM Symposium on Applied Computing, Track on Advances in Spatial and Image-based Information Systems*, pages 881–887, Sierre, Switzerland, 2010. (Cited on page 59.)

[DSL05] Chabane Djeraba, Nicu Sebe, and Michael S. Lew. Systems and architectures for multimedia information retrieval. *Multimedia Systems*, 10(6):457–463, 2005. (Cited on page 49.)

[DTG+08] Mario Döller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian S Burnett. The MPEG Query Format: On the way to unify the access to multimedia retrieval systems. *IEEE Multimedia*, 15(4):82–95, 2008. (Cited on pages 53 and 79.)

[Dun03] Lynne Dunckley. *Multimedia Databases: An Object Relational Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003. (Cited on pages 22 and 25.)

[EABC+11] Amr El Abaddi, Lars Backstrom, Soumen Chakrabarti, Alejandros Jaimes, Jure Leskovec, and Andrew Tomkins. Social media: Source of information or bunch of noise. In *Proceedings of the 20th International Conference on World Wide Web*, pages 327–328, New York, NY, USA, 2011. ACM. (Cited on pages 3 and 16.)

[EB11] Jean-Pierre Evain and Tobias Bürger. Semantic web, linked data and broadcasting – more in common than you'd think! *EBU Technical Review*, 2011(Q1):1–13, 2011. (Cited on page 62.)

[EHSM08] Hugo Jair Escalante, Carlos A. Hérnadez, Luis Enrique Sucar, and Manuel Montes. Late fusion of heterogeneous methods for multimedia image retrieval. In *Proceedings of the 1 st ACM International Conference on Multimedia Information Retrieval*, pages 172–179, 2008. (Cited on page 111.)

[Eid03] Horst Eidenberger. Distance measures for MPEG-7-based retrieval. In *Proceedings of the 5th ACM International Workshop on Multimedia Information Retrieval*, pages 130–137, New York, NY, USA, 2003. ACM. (Cited on page 42.)

[FE72] Leonard Fisher and Donald Elchesen. Effectiveness of combining title words and index terms in machine retrieval searches. *Nature*, (238):109–110, 1972. (Cited on page 110.)

[FM81] K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3 – 16, 1981. (Cited on page 39.)

[FSA+95] Myron Flickner, Harpreet S. Sawhney, Jonathan 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. (Cited on page 40.)

[FW04] David Fallside and Priscilla Walmsley. XML Schema Part 0: Primer Second Edition. W3C Recommendation. 28 October, 2004. http://www.w3.org/TR/xmlschema-0/. (Cited on page 28.)

[GC05] R. Garcia and O. Celma. Semantic integration and retrieval of multimedia metadata. In *Proceedings of 4rd Workshop on Knowledge Markup and Semantic Annotation, colocated to the International Semantic Web Conference*, pages 69–80, 2005. (Cited on pages 78 and 95.)

[GG98] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Survey*, 30(2):170–231, 1998. (Cited on pages 44 and 45.)

[GN08] P. Geetha and Vasumathi Narayanan. A survey of content-based video retrieval. *Journal of Computer Science*, 4(6):474–486, 2008. (Cited on page 6.)

[GNW95] Michel Grabisch, Hung Nguyen, and Elbert Walker. *Fundamentals of uncertainty calculi, with applications to fuzzy inference*. Kluwer Academic, Dordrecht, 1995. (Cited on pages 112 and 141.)

[Gom11] Hasssan Gomaa. *Software modeling and design - UML, use cases, patterns, and software architectures*. Cambridge University Press, 2011. (Cited on page 52.)

[Gra94a] G. Graefe. Volcano: An extensible and parallel query evaluation system. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):120–135, Februar 1994. (Cited on page 101.)

[Gra94b] Goetz Graefe. Volcano - an extensible and parallel query evaluation system. *IEEE Transactions on Knowledge Data Engineering*, 6(1):120–135, 1994. (Cited on page 118.)

[Gut84] Antonin Guttman. R-Trees: A dynamic index structure for spatial searching. In *Proceedings of the International Conference on Management of Data*, pages 47–57. ACM Press, 1984. (Cited on page 45.)

[GW01] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. (Cited on page 40.)

[Har95] Donna Harman. Overview of the third Text REtrieval Conference. In *Proceedings of the 3rd Text REtrieval Conference*, pages 1–19. NIST Special Publication 500-207, 1995. (Cited on page 13.)

[Hau07] Michael Hausenblas. Multimedia Vocabularies on the Semantic Web. W3C Incubator Group Report. 24 July, 2007. http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/. (Cited on page 27.)

[HBNM11] Martin Höffernig, Werner Bailer, Günter Nagler, and Helmut Mülner. Mapping audiovisual metadata formats using formal semantics. In *Proceedings of the 5th Conference on Semantic and Digital Media Technology*, volume 6725 of *Lecture Notes in Computer Science*, pages 80–94. 2011. (Cited on pages 71 and 72.)

[Hen03] Peter A. Henning. *Taschenbuch Multimedia*. Carl Hanser Verlag, München, Germany, 2003. (Cited on page 11.)

[HK10] Bernhard Haslhofer and Wolfgang Klas. A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys*, 42(2):7:1–7:37, March 2010. (Cited on pages 23 and 24.)

[HKP+09] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer. W3C Recommendation. 27 October, 2009. http://www.w3.org/TR/owl2-primer/. (Cited on page 31.)

[HKRS08] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, and York Sure. *Semantic Web*. Springer-Verlag Berlin Heidelberg, 2008. (Cited on page 29.)

[HLMS08] Alan Hanjalic, Rainer Lienhart, Wei-Ying Ma, and John R. Smith. The holy grail of multimedia information retrieval: So close or yet so far away? *IEEE Multimedia*, 96(4):541 –547, April 2008. (Cited on page 17.)

[HM98] K. Holtman and A. Mutz. Transparent Content Negotiation in HTTP. RFC 2295 (Experimental), March 1998. (Cited on page 30.)

[HON+08] Lynda Hardman, Zeljko Obrenovic, Frank Nack, Brigitte Kerhervé, and Kurt W. Piersol. Canonical processes of semantically annotated media production. *Multimedia Systems*, 14(6):327–340, 2008. (Cited on pages 18 and 34.)

[HR96] Stacie Hibino and Elke A. Rundensteiner. A visual multimedia query language for temporal analysis of video data. In *Multimedia Database Systems*, pages 123–159. 1996. (Cited on page 52.)

[HS85] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100 – 132, 1985. (Cited on page 39.)

[HSA05] Stavros Harizopoulos, Vladislav Shkapenyuk, and Anastassia Ailamaki. QPipe: a simultaneously pipelined relational query engine. In *Proceedings of the ACM International Conference on Management of Data*, pages 383–394, New York, NY, USA, 2005. ACM. (Cited on page 106.)

[HSD11] Jonathon S. Hare, Sina Samangooei, and David P. Dupplaw. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th Conference on Multimedia*, pages 691–694, 2011. (Cited on page 128.)

[HSLZ11] Zi Huang, Heng Tao Shen, Jiajun Liu, and Xiaofang Zhou. Effective data co-reduction for multimedia similarity search. In *Proceedings of the SIGMOD/PODS Conference*, pages 1021–1032. ACM, 2011. (Cited on page 47.)

[HSS+08] Zi Huang, Heng Tao Shen, Jie Shao, Stefan M. Rüger, and Xiaofang Zhou. Locality condensation: A new dimensionality reduction method for image retrieval. In *Proceedings of the ACM Conference on Multimedia*, pages 219–228. ACM, 2008. (Cited on page 47.)

[HWZ02] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. In *Proceedings of the 17th Symposium on Logic in Computer Science*, pages 393–402, Washington, DC, USA, 2002. (Cited on page 30.)

[ISO86] ISO - International Organization for Standardization. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML). ISO 8879, 1986. `http://www.iso.org/iso/catalogue_detail.htm?csnumber=16387`. (Cited on page 27.)

[Jae05] Bernd Jaehne. *Digital Image Processing*. Springer, 6th edition edition, 2005. (Cited on page 38.)

[Jai08] Ramesh Jain. Multimedia information retrieval: watershed events. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 229–236, New York, NY, USA, 2008. ACM. (Cited on page 19.)

[JCH04] Yiming Ji, Kai H. Chang, and Chi-Cheng Hung. Efficient edge detection and object segmentation using Gabor filters. In *Proceedings of the 42nd annual Southeast regional conference*, pages 454–459, New York, NY, USA, 2004. ACM. (Cited on page 41.)

[JJBC01] Corinne Jörgensen, Alejandro Jaimes, Ana B. Benitez, and Shih-Fu Chang. A conceptual framework and empirical research for classifying visual descriptors. *Journal of the American Society for Information Science and Technology*, 52(11):938–947, 2001. (Cited on page 39.)

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446, October 2002. (Cited on page 120.)

[JLST01] Hosagrahar Visvesvaraya Jagadish, Laks V. S. Lakshmanan, Divesh Srivastava, and Keith Thompson. TAX: A tree algebra for XML. In *Proceedings of the 8th Workshop on Databases and Programming Languages*, pages 149–164, 2001. (Cited on page 53.)

[JS06] Bernard J. Jansen and Amanda Spink. How are we searching the world wide web? a comparison of nine search engine transaction logs. *Information Processing and Management*, 42:248–263, January 2006. (Cited on page 137.)

[KBD+05] Harald Kosch, Laszlo Böszörmanyi, Mario Döller, Mulugeta Libsie, Peter Schojer, and Andrea Kofler. The life cycle of multimedia metadata. *IEEE Multimedia*, 12:80–86, 2005. (Cited on pages 18 and 33.)

[KC04] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. 10 February, 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. (Cited on page 29.)

[Kos00] Donald Kossmann. The state of the art in distributed query processing. *ACM Computing Survey*, 32:422–469, December 2000. (Cited on page 90.)

[KS97] Norio Katayama and Shin'ichi Satoh. The SR-Tree: An index structure for high-dimensional nearest neighbor queries. In *Proceedings of the International Conference on Management of Data*, pages 369–380. ACM Press, 1997. (Cited on page 47.)

[KSB11] Thomas Kurz, Sebastian Schaffert, and Tobias Bürger. LMF – a framework for linked media. In *Proceedings of the Workshop on Multimedia on the Web collocated to i-KNOW/i-SEMANTICS*, pages 1–4, September 2011. (Cited on pages 71 and 72.)

[KY95] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1st edition, 1995. (Cited on page 141.)

[LBB+12] WonSuk Lee, Werner Bailer, Tobias Bürger, Pierre-Antoine Champin, Jean-Pierre Evain, Véronique Malaisé, Thierry Michel, Felix Sasaki, Joakim Söderberg, Florian Stegmaier, and John Strassner. Ontology for media resources 1.0. W3C Recommendation. 09 February, 2012. http://www.w3.org/TR/2012/REC-mediaont-10-20120209/. (Cited on page 61.)

[LBEK02] Jobst Löffler, Konstantin Biatov, Christian Eckes, and Joachim Köhler. IFINDER: An MPEG-7-based retrieval system for distributed multimedia content. In *ACM Multimedia*, pages 431–435, 2002. (Cited on pages 77 and 95.)

[LC86] Tobin J. Lehman and Michael J. Carey. A study of index structures for main memory database management systems. In *Proceedings of the 12th International Conference on Very Large Data Bases*, pages 294–303, 1986. (Cited on page 45.)

[LC05] Kuen-Long Lee and Ling-Hwei Chen. An efficient computation method for the texture browsing descriptor of MPEG-7. *Image and Vision Computing*, 23(5):479–489, 2005. (Cited on page 41.)

[LC06] Chia-Han Lin and A. L. P. Chen. Indexing and matching multiple-attribute strings for efficient multimedia query processing. *IEEE Transactions on Multimedia*, 8(2):408–411, 2006. (Cited on page 99.)

[LCH01] Peiya Lui, Amit Charkraborty, and Liang H. Hsu. A logic approach for MPEG-7 XML document queries. In *Proceedings of the Extreme Markup Languages*, pages 1–15, 2001. (Cited on page 52.)

[LCS97] Dik L. Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *Software, IEEE*, 14(2):67 –75, March/April 1997. (Cited on pages 110 and 111.)

[LDMW11] Wim Van Lancker, Davy Van Deursen, Erik Mannens, and Rik Van de Walle. Harmonizing media annotations and media fragments. In *Proceedings of the Workshop on Multimedia on the Web collocated to i-KNOW/i-SEMANTICS*, pages 1–4, September 2011. (Cited on pages 71 and 72.)

[LJA11] Herwig Lejsek, Björn P. Jónsson, and Laurent Amsaleg. NV-Tree: Nearest neighbors at the billion scale. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 54:1–54:8. ACM, 2011. (Cited on page 48.)

[LKKL93] Joon Ho Lee, Won Yong Kin, Myoung Ho Kim, and Yoon Joon Lee. On the evaluation of Boolean operators in the extended Boolean retrieval framework. In *Proceedings of the 16th International ACM Conference on Research and Development in Information Retrieval*, pages 291–297, New York, NY, USA, 1993. ACM. (Cited on page 15.)

[LMH$^+$85] Guy M. Lohman, C. Mohan, Laura M. Haas, Dean Daniels, Bruce G. Lindsay, Patricia G. Selinger, and Paul F. Wilms. Query processing in R*. In *Query Processing in Database Systems*, pages 31–47. Springer, 1985. (Cited on page 90.)

[LMS10] Sébastien Laborie, Ana-Maria Manzat, and Florence Sèdes. A generic framework for the integration of heterogeneous metadata standards into a multimedia information retrieval system. In *Proceedings of the Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 80–83, 2010. (Cited on pages 78 and 95.)

[LR05] Bin Liu and E. Rundensteiner. Revisiting pipelined parallelism in multi-join query processing. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 829–840. VLDB Endowment, 2005. (Cited on page 102.)

[LSDJ06] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, 2006. (Cited on page 50.)

[LW03] Jia Li and James Ze Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003. (Cited on page 130.)

[MA02] Mark Montague and Javed Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the 11 th international conference on Information and knowledge management*, pages 538–548, 2002. (Cited on pages 52, 110 and 111.)

[ME01] Jim Melton and Andrew Eisenberg. SQL multimedia and application packages (SQL/MM). *SIGMOD Record*, 30(4):97–102, 2001. (Cited on page 52.)

[MGCB08] Mauricio Marin, Veronica Gil-Costa, and Carolina Bonacic. A search engine index for multimedia content. In *Proceedings of the 14th International Euro-Par Conference on Parallel Processing*, pages 866–875, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 100.)

[MH04] Deborah McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation. 10 February, 2004. `http://www.w3.org/TR/owl-features/`. (Cited on page 30.)

[Mit06] Ankush Mittal. An overview of multimedia content-based retrieval strategies. *Informatica*, 30:347–356, 2006. (Cited on page 14.)

[MPR+99] Angelo Morzenti, Matteo Pradella, Matteo Rossi, Stefano Russo, and Antonio Sergio. A case study in object-oriented modeling and design of distributed multimedia applications. In *International Symposium on Software Engineering for Parallel and Distributed Systems*, pages 217–223, 1999. (Cited on page 21.)

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, Cambridge, England, 2008. (Cited on pages 12, 14 and 119.)

[MS07] Manuel Möller and Michael Sintek. A Generic Framework for Semantic Medical Image Retrieval. In *Proceedings of the 1st Workshop of*

Knowledge Acquisition from Multimedia Content, volume 253, pages 18–32, Genova, Italy, 2007. (Cited on pages 77 and 95.)

[MSGA13] Diana Moise, Denis Shestakov, Gylfi Gudmundsson, and Laurent Amsaleg. Indexing and searching 100m images with map-reduce. In *Proceedings of the 3rd ACM International Conference on Multimedia Retrieval*, pages 17–24, New York, NY, USA, 2013. ACM. (Cited on page 47.)

[MSMV07] Jorge Manjarrez-Sanchez, J. Martinez, and Patrick Valduriez. A data allocation method for efficient content-based retrieval in parallel multimedia databases. In *Frontiers of High Performance Computing and Networking*, volume 4743 of *Lecture Notes in Computer Science*, pages 285–294. Springer Berlin / Heidelberg, 2007. (Cited on page 100.)

[MTD03] Danilo Montesi, Alberto Trombetta, and Peter A. Dearnley. A similarity based relational algebra for web and multimedia data. *Journal of Information Processing and Management: Modelling vagueness and subjectivity in information*, 39(2):307–322, March 2003. (Cited on page 53.)

[MW03] Klaus Meyer-Wegener. *Multimediale Datenbanken: Einsatz von Datenbanktechnik in Multimedia-Systemen*. Leitfäden der Informatik. B. G. Teubner Verlag, Wiesbaden, Germany, 2003. (Cited on page 11.)

[Nac00] Frank Nack. All content counts: The future in digital media computing is meta. *IEEE Multimedia*, 7(3):10–13, 2000. (Cited on pages 21 and 33.)

[Nat04] National Information Standards Organization. Understanding metadata. NISO Press, 2004. http://www.niso.org/publications/press/UnderstandingMetadata.pdf. (Cited on page 12.)

[Neu11] Thomas Neumann. Efficiently compiling efficient query plans for modern hardware. *Proceedings of the VLDB Endowment*, 4(9):539–550, 2011. (Cited on page 103.)

[NTX+07] Apostol! Natsev, Jelena Tešić, Lexing Xie, Rong Yan, and John R. Smith. Ibm multimedia search and retrieval system. In *Proceedings of the 6th International Conference on Image and Video Retrieval*, pages 645–645, 2007. (Cited on page 50.)

[OMG11] Object Management Group OMG. Meta Object Facility (MOF) Core Specification. Version 2.4.1, August 2011. http://www.omg.org/spec/MOF/2.4.1/PDF/. (Cited on page 26.)

[ONH04] Jacco van Ossenbruggen, Frank Nack, and Lynda Hardman. That obscure object of desire: Multimedia metadata on the web, part 1. *IEEE Multimedia*, 11:38–48, 2004. (Cited on pages 22 and 34.)

[Pep02] Steve Pepper. The TAO of Topic Maps: Finding the Way in the Age of Infoglut. http://www.ontopia.net/topicmaps/materials/tao.html, April 2002. (Cited on page 31.)

[PG08] Jeremy Pickens and Gene Golovchinsky. Ranked feature fusion models for ad hoc retrieval. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 893–900. ACM, 2008. (Cited on page 48.)

[PMMdW09] Chris Poppe, Gaëtan Martens, Erik Mannens, and Rik Van de Walle. Personal content management system: A semantic approach. *Journal of Visual Communication and Image Representation*, 20(2):131–144, 2009. (Cited on page 59.)

[Poe06] Iman Poernomo. The meta-object facility typed. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1845–1849. ACM, 2006. (Cited on page 26.)

[PP93] Nikhil R. Pal and Sankar K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. (Cited on page 41.)

[Pra97] B. Prabhakaran. *Multimedia Database Management Systems*. Kluwer Academic Publishers, Boston, 1997. (Cited on page 37.)

[PS08] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. W3C Recommendation. 15 January, 2008. `http://www.w3.org/TR/rdf-sparql-query/`. (Cited on page 53.)

[Rah94] Erhard Rahm. *Mehrrechner–Datenbanksysteme: Grundlagen der verteilten und parallelen Datenbankverwaltung*. Addison-Wesley, 1994. (Cited on page 51.)

[RHW10] R.M. Rasli, Su-Cheng Haw, and Chee-Onn Wong. A survey on optimizing video and audio query retrieval in multimedia databases. In *Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering*, volume 2, pages V2–302 –V2–306, August 2010. (Cited on page 19.)

[RM12] Miriam Redi and Bernard Merialdo. Exploring two spaces with one feature: Kernelized multidimensional modeling of visual alphabets. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pages 20:1–20:8. ACM, 2012. (Cited on page 47.)

[Rob03] Stephen Robertson. The unified model revisited. In *Proceedings of the Workshop on Mathematical / Formal Models in Information Retrieval, colocated to ACM SIGIR Conference*, pages 1–11, 2003. (Cited on page 3.)

[Rud11] Sebastian Rudolph. Foundations of description logics. In *Reasoning Web. Semantic Technologies for the Web of Data*, volume 6848 of *Lecture Notes in Computer Science*, pages 76–136. 2011. (Cited on page 31.)

[Rue10] Stefan Rueger. *Multimedia information retrieval*. Synthesis Lectures on Information Concepts, Retrieval and Services. Morgan & Claypool Publishers, 2010. (Cited on pages 16 and 41.)

[Sag94] Hans Sagan. *Space-Filling Curves*. Springer, 1 edition, September 1994. (Cited on page 48.)

[Sam10] Hanan Samet. Techniques for similarity searching in multimedia databases. *PVLDB*, 3(2):1649–1650, 2010. (Cited on page 47.)

[San06] Simone Santini. Data modeling, multimedia. In Borko Furht, editor, *Encyclopedia of Multimedia*, pages 149–154. Springer US, 2006. (Cited on page 21.)

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24:513–523, August 1988. (Cited on page 14.)

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991. (Cited on page 40.)

[SBB+09] Florian Stegmaier, Werner Bailer, Tobias Bürger, Mario Döller, Martin Höffernig, Wonsuk Lee, Véronique Malaisé, Chris Poppe, Raphael Troncy, Harald Kosch, and Rik Van de Walle. How to align media metadata schemas? Design and implementation of the media ontology. In *Proceedings of the 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies in conjunction with SAMT*, volume 539, pages 56–69, Graz, Austria, December 2009. (Cited on pages 57 and 65.)

[SBB+13] Florian Stegmaier, Werner Bailer, Tobias Burger, Mari Carmen Suarez-Figueroa, Erik Mannens, Jean-Pierre Evain, Martin Höffernig, Pierre-Antoine Champin, Mario Döller, and Harald Kosch. Unified access to media metadata on the web. *IEEE Multimedia*, 20(2):22–29, 2013. (Cited on pages 57 and 71.)

[SBH06] Alf-Christian Schering, Ammar S. Balouch, and Andreas Heuer. BSA-Algebra für XQuery, Operation - Optimierungsregeln und Anwendungen. In *Proceedings of the 18^{t}h GI-Workshop on the Foundations of Databases*, pages 135–139, 2006. (Cited on page 53.)

[SBH+13] Florian Stegmaier, Werner Bailer, Martin Höffernig, WonSuk Lee, and Chris Poppe. API for media resources 1.0. W3C Proposed Recommendation. 23 July, 2013. http://www.w3.org/2008/WebVideo/Annotations/drafts/API10/PR2/. (Cited on page 61.)

[Sch06] Ingo Schmitt. *Ähnlichkeitssuche in Multimedia-Datenbanken: Retrieval, Suchalgorithmen und Anfrageverarbeitung*. Oldenbourg Wissenschaftsverlag GmbH, Munich, Germany, 2006. (Cited on pages 11, 12 and 15.)

[Sch08] Ingo Schmitt. QQL: A DB&IR query language. *VLDB Journal*, 17(1):39–56, 2008. (Cited on page 53.)

[SDK+10] Florian Stegmaier, Mario Döller, Harald Kosch, Andreas Hutter, and Thomas Riegel. AIR: Architecture for interoperable retrieval on distributed and heterogeneous multimedia repositories. In *The 11th Workshop on Image Analysis for Multimedia*, pages 1–4, 2010. (Cited on page 77.)

[SDS+11] Florian Stegmaier, Mario Döller, Kai Schlegel, Harald Kosch, Sascha Seifert, Martin Kramer, Thomas Riegel, Andreas Hutter, Marisa Thoma, Hans-Peter Kriegel, Matthias Hammon, and Alexander Cavallaro. Generische Datenintegration zur semantischen Diagnoseunterstützung im Projekt THESEUS: MEDICO. In *Proceedings of the Workshop on Datenmanagement und Interoperabilität im Gesundheitswesen, co-located with the 41th Conference of the Gesellschaft für Informatik e.V.*, pages 1–14, 2011. (Cited on page 96.)

[Sei98] Thomas Seidl. *Adaptable Similarity Search in 3-D Spatial Database Systems*. Herbert Utz Verlag Wissenschaft, Munich, Germany, 1998. (Cited on page 42.)

[SES12] Ansgar Scherp, Daniel Eißing, and Carsten Saathoff. A method for integrating multimedia metadata standards and metadata formats with the Multimedia Metadata Ontology. *International Journal on Semantic Computing*, Accepted for Publication August 2011, in print for 2012. (Cited on pages 71 and 72.)

[SF94] Joseph Shaw and Edward Fox. Combination of multiple searches. In *Proceedings of the 2 nd Text Retrieval Conference*, pages 243–252, 1994. (Cited on pages 90 and 111.)

[SFAC11] Mari Carmen Suàrez-Figueroa, Ghislain Auguste Atemezing, and Oscar Corcho. The landscape of multimedia ontologies in the last decade. *Multimedia Tools and Applications*, pages 1–23, 2011. (Cited on page 35.)

[SGD+09] Florian Stegmaier, Udo Gröbner, Mario Döller, Harald Kosch, and Gero Baese. Evaluation of current RDF database solutions. In *Workshop on Semantic Multimedia Database Technologies, co-located with the 4th International Conference on Semantic and Digital Media Technologies*, volume 539, pages 1–14. CEUR-WS.org, 2009. (Cited on page 29.)

[SGN+11] Xingzhi Sun, Leiguang Gong, Apostol Natsev, Xiaofei Teng, Li Tian, Tao Wang, and Yue Pan. Image modality classification: a late fusion method based on confidence indicator and closeness matrix. In *Proceedings of the 1 st ACM International Conference on Multimedia Retrieval*, pages 55:1–55:7, 2011. (Cited on page 111.)

[Sik01] Thomas Sikora. The MPEG-7 visual standard for content description - an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696 –702, 2001. (Cited on page 40.)

[Sin01] Amit Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001. (Cited on page 12.)

[SKS10] Michael Springmann, Dietmar Kopp, and Heiko Schuldt. QbS: Searching for known images using user-drawn sketches. In *Proceedings of the 1st International Conference on Multimedia Information Retrieval*, pages 417–420, New York, NY, USA, 2010. ACM. (Cited on page 16.)

[SL90] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Survey*, 22(3):183–236, 1990. (Cited on page 51.)

[SLP11] Franco M. Segarra, Luis A. Leiva, and Roberto Paredes. A relevant image search engine with late fusion: mixing the roles of textual and visual descriptors. In *Proceedings of the 16 th International Conference on Intelligent User Interfaces*, pages 455–456, 2011. (Cited on page 111.)

[Smi08] John R. Smith. The search for interoperability. *IEEE Multimedia*, 15:84–87, 2008. (Cited on page 4.)

[SOZ05] Heng Tao Shen, Beng Chin Ooi, and Xiaofang Zhou. Towards effective indexing for very large video sequence database. In *Proceedings of the*

International Conference on Management of Data, pages 730–741. ACM, 2005. (Cited on page 47.)

[SRF87] Timos K. Sellis, Nick Roussopoulos, and Christos Faloutsos. The R^+-Tree: A dynamic index for multi-dimensional objects. In *Proceedings of* 13^{th} *International Conference on Very Large Data Bases*, pages 507–518. Morgan Kaufmann, 1987. (Cited on page 46.)

[SS04a] Gerald Schaefer and Michal Stich. UCID - An uncompressed colour image database. In *In Storage and Retrieval Methods and Applications for Multimedia 2004, volume 5307 of Proceedings of SPIE*, pages 472–480, 2004. (Cited on page 130.)

[SS04b] Ingo Schmitt and Nadine Schulz. Similarity relational calculus and its reduction to a similarity algebra. In *Proceedings of the* 7^{th} *Symposium on Foundations of Information and Knowledge Systems*, pages 252–272, 2004. (Cited on page 53.)

[SS06] John R. Smith and Peter Schirling. Metadata standards roundup. *IEEE Multimedia*, 13(2):84–88, 2006. (Cited on pages 18, 26 and 33.)

[SSB+12] Florian Stegmaier, Kai Schlegel, Sebastian Bayerl, Mario Döller, and Harald Kosch. Optimization of federated multimedia queries in an external meta-search engine. In *The* 1^{st} *Workshop on Multimedia Databases and Data Engineering, co-located with the* 38^{th} *Conference on Very Large Databases*, pages 1–8, 2012. (Cited on pages 77, 99 and 122.)

[SSH05] Ingo Schmitt, Nadine Schulz, and Thomas Herstel. WS-QBE: A QBE-Like query language for complex multimedia queries. In *Proceedings of the Eleventh International Multi-Media Modelling Conference*, pages 222–229. IEEE Computer Society, 2005. (Cited on page 53.)

[ST07] Nicu Sebe and Qi Tian. Personalized multimedia retrieval: the new trend? In *Proceedings of the* 9^{th} *International Workshop on Multimedia Information Retrieval*, pages 299–306, New York, NY, USA, 2007. ACM. (Cited on page 16.)

[Ste99] Ralf Steinmetz. *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*. Springer, Berlin, Germany, 1999. (Cited on page 11.)

[Ste10] Florian Stegmaier. Interoperable and unified multimedia retrieval in distributed and heterogeneous environments. In *Proceedings of the International Conference on Multimedia*, pages 1705–1706, 2010. (Cited on page 5.)

[Str09] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge, 4^{th} edition, 2009. (Cited on page 15.)

[STS+11] Sascha Seifert, Marisa Thoma, Florian Stegmaier, Matthias Hammon, Martin Kramer, Martin Huber, Hans-Peter Kriegel, Alexander Cavallaro, and Dorin Comaniciu. Combined semantic and similarity search in medical image databases. In *Proceedings of the SPIE Medical Imaging Conference 2011: Advanced PACS-based Imaging Informatics and Therapeutic Applications, Lake Buena Vista, FL, USA*, volume 7967, page 796702, 2011. (Cited on page 96.)

[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 Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000. (Cited on page 17.)

[SWS05] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13 th ACM International Conference on Multimedia*, pages 399–402, 2005. (Cited on page 111.)

[SWY75] Gerard Salton, Andrew Wong, and Chung-Shuh Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620, November 1975. (Cited on page 15.)

[TC92] Gabriel Taubin and David B. Cooper. Geometric invariance in computer vision. chapter Object recognition based on moment (or algebraic) invariants, pages 375–397. MIT Press, Cambridge, MA, USA, 1992. (Cited on page 41.)

[TCL+07] R. Troncy, Ò. Celma, S. Little, R. Garcia, and C. Tsinaraki. MPEG-7 based multimedia ontologies: Interoperability support or interoperability issue? In *1st Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies*, Genova, Italy, 2007. (Cited on page 35.)

[TD09] Ruben Tous and Jaime Delgado. A LEGO-like Metadata Architecture for Image Search&Retrieval. In *Proceedings of the 3rd Workshop on Multimedia Data Mining and Management*, pages 246–250, 2009. (Cited on pages 77 and 95.)

[TDMR+05] V. Tzouvaras, S. Dasiopoulou, F. Martin-Recuerda, G. Stoilos, Y. Kompatsiaris, and G. Stamou. Multimedia analysis and annotation requirements for the semantic web. In *2nd European Workshop on the Integration of Knowledge, held in conjunction with Semantics and Digital Media Technology*, pages 443 –450, 2005. (Cited on page 35.)

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, July 2008. (Cited on page 42.)

[TMF10] Raphaël Troncy, Bartosz Malocha, and André T. S. Fialho. Linking events with media. In *Proceedings of the 6th Conference on Semantic Systems*, pages 42:1–42:4, 2010. (Cited on page 72.)

[TMPD12] Raphaël Troncy, Erik Mannens, Silvia Pfeiffer, and Davy van Deursen. Media Fragments URI 1.0 (basic). W3C Proposed Recommendation. 15 March, 2012. `http://www.w3.org/TR/2012/PR-media-frags-20120315/`. (Cited on pages 39 and 149.)

[TN11] Hung-Khoon Tan and Chong-Wah Ngo. Fusing heterogeneous modalities for video and image re-ranking. In *Proceedings of the 1 st ACM International Conference on Multimedia Retrieval*, pages 15:1–15:8, 2011. (Cited on page 111.)

[TWV05] Rainer Typke, Frans Wiering, and Remco C. Veltkamp. A survey of music information retrieval systems. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 153–160, London, UK, 2005. Queen Mary, University of London. (Cited on page 6.)

[UDJ08] Thierry Urruty, Chabane Djeraba, and Joemon M. Jose. An efficient indexing structure for multimedia data. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, pages 313–320. ACM, 2008. (Cited on page 47.)

[UDUG04] G. Ünel, M. E. Dönderler, Ulusoy, and U. Güdkbay. An efficient query optimization strategy for spatio-temporal queries in video databases. *Journal of Systems and Software*, 73(1):113–131, September 2004. (Cited on page 99.)

[VCPF08] Eduardo Valle, Matthieu Cord, and Sylvie Philipp-Foliguet. High-dimensional descriptor indexing for large multimedia databases. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 739–748, New York, NY, USA, 2008. ACM. (Cited on page 47.)

[VH98] Ellen Marie Voorhees and Donna Harman. Overview of the sixth Text REtrieval Conference (TREC-6). In *NIST Special Publication 500-240: The Sixth Text REtrieval Conference (TREC-6)*, pages 1–24, 1998. (Cited on page 129.)

[WBDB$^+$06] James Z. Wang, Nozha Boujemaa, Alberto Del Bimbo, Donald Geman, Alexander G. Hauptmann, and Jelena Tesić. Diversity in multimedia information retrieval research. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 5–12, New York, NY, USA, 2006. ACM. (Cited on page 16.)

[WCW11] Zongda Wu, Zhongsheng Cao, and Yuanzhen Wang. Multimedia selection operation placement. *Multimedia Tools and Applications*, 54:69–96, 2011. (Cited on page 99.)

[WJ96] David A. White and Ramesh Jain. Similarity indexing with the SS-Tree. In *Proceedings of the International IEEE Conference on Data Engineering*, pages 516–523. IEEE Computer Society, 1996. (Cited on page 46.)

[WK10] Xiangyu Wang and Mohan Kankanhalli. Portfolio theory of multimedia fusion. In *Proceedings of the 18 st ACM International Conference on Multimedia*, pages 723–726, 2010. (Cited on page 111.)

[WLLC10] Zongda Wu, Chenglang Lu, Jianfeng Lu, and Zhongsheng Cao. Mrea: A relation-extension algebra for processing umql-based multimedia queries. In *Proceedings of the 3rd Biomedical Engineering and Informatics*, volume 7, pages 2692–2697, 2010. (Cited on page 53.)

[WP11] Li Weng and Bart Preneel. Image distortion estimation by hash comparison. In *Proceedings of the 17th International Conference on MultimediaÂ Modeling*, pages 62–72, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on page 105.)

[WSF10] Peter Wilkins, Alan F. Smeaton, and Paul Ferguson. Properties of optimally weighted data fusion in cbmir. In *Proceeding of the 33 rd International Conference on Research and Development in Information Retrieval*, pages 643–650, 2010. (Cited on page 111.)

[WWL+13] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of nDCG type ranking measures. *CoRR*, abs/1304.6480, 2013. (Cited on page 121.)

[WYLD10] Brandyn White, Tom Yeh, Jimmy Lin, and Larry Davis. Web-scale computer vision using MapReduce for multimedia data mining. In *Proceedings of the 10th Workshop on Multimedia Data Mining*, pages 9:1–9:10, 2010. (Cited on page 47.)

[XXE07] Guangming Xing, Zhonghang Xia, and Andrew Ernest. Building automatic mapping between XML documents using approximate tree matching. In *Proceedings of Symposium on Applied Computing*, pages 525–526, 2007. (Cited on page 58.)

[Yer03] F. Yergeau. UTF-8, a transformation format of ISO 10646. RFC 3629 (Standard), November 2003. (Cited on page 28.)

[YFM+09] Rong Yan, Marc-Olivier Fleury, Michele Merler, Apostol Natsev, and John R. Smith. Large-scale multimedia semantic concept modeling using robust subspace bagging and MapReduce. In *Proceedings of*

1st Workshop on Large-scale Multimedia Retrieval and Mining, pages 35–42, 2009. (Cited on page 47.)

[YH03] Rong Yan and Alexander Hauptmann. The combination limit in multimedia retrieval. In *Proceedings of the 11 th ACM International Conference on Multimedia*, pages 339–342, 2003. (Cited on page 111.)

[YHJ03] Rong Yan, Alexander Hauptmann, and Rong Jin. Multimedia search with pseudo-relevance feedback. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 238–247, 2003. (Cited on page 111.)

[YLL+03] Xia Yang, MongLi Lee, Tok Wang Ling, Lee Tok, and Wang Ling. Resolving structural conflicts in the integration of XML schemas: A semantic approach. In *Proceedings of the International Conference on Conceptual Modeling*, pages 520–533, 2003. (Cited on page 59.)

[YYH04] Rong Yan, Jun Yang, and Alexander G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12 th ACM International Conference on Multimedia*, pages 548–555, 2004. (Cited on page 111.)

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. (Cited on page 141.)

[ZBB+12] David Zellhöfer, Maria Bertram, Thomas Böttcher, Christoph Schmidt, Claudius Tillmann, and Ingo Schmitt. Pythiasearch: a multiple search strategy-supportive multimedia retrieval system. In *Proceedings of the 2nd International Conference on Multimedia Retrieval*, pages 59:1–59:2, 2012. (Cited on pages 50 and 128.)

[ZIL12] Dengsheng Zhang, Md. Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012. (Cited on page 40.)

[ZW04] Sonja Zillner and Werner Winiwarter. Integrating ontology knowledge into a query algebra for multimedia meta objects. In *Web Information Systems Engineering*, volume 3306 of *Lecture Notes in Computer Science*, pages 629–640, 2004. (Cited on page 53.)