MASARYK UNIVERSITY
FACULTY OF INFORMATICS

Towards Large-Scale Multi-Modal Image Search

PH.D. THESIS

Petra Budíková

Supervisor: prof. Ing. Pavel Zezula, CSc.

Brno, 2013

Declaration

I hereby declare that this dissertation is my own original work and has not been submitted before to any institution for assessment purposes. All sources, references and literature used during elaboration of this work are properly cited and listed in complete reference to the due source.

Petra Budíková

Acknowledgements

I have never built a house, I have never planted a forest. Writing the dissertation thesis was one of the most demanding tasks of my life, and I wouldn't have done it alone. Therefore, I would like to thank all who supported me on the way, in particular

- to my supervisor, professor Pavel Zezula, for overall guidance, friendly approach and well-placed criticism that inspired me to improve my work;
- to my consultant, doctor Michal Batko, for his patience, countless pieces of good advice, and encouragement it weak moments;
- to my husband Pavel – for everything, but in particular for not throwing me and my computer out of the window during the last few months;
- to my parents, for lifelong love and support;
- to my friend Tereza Madronová, for improving the language quality of this thesis (all errors left there are my work, not hers);
- and to all my colleagues and friends who patiently participated in the experiments necessary for my research.

Abstract

In the modern information society, we experience an unprecedented explosion of digital information. Fueled by the easy availability of acquisition and storing technology, the amounts of digital data as well as its complexity continue to grow at a high speed. This situation offers many opportunities for information retrieval, but also presents huge challenges in the area of data management.

Due to the high internal complexity of many modern data types such as multimedia, traditional database systems that exploit exact match paradigm are no longer suitable for organization and retrieval of such data. The similarity-based approach, which has attracted a lot of research attention in the last two decades, represents a promising alternative, as the utilization of similarity is natural to human cognition and learning. However, the concept of similarity has been shown to be difficult to define and implement, and the similarity-based retrieval is still far from becoming a mature technology. The research efforts proceed on many levels, including psychological studies of similarity, analysis of multimedia properties, and development of data indexing and retrieval techniques. It is also becoming clear that the concept of similarity is subjective, context-dependent and multi-faceted, which needs to be reflected by the data management tools. Therefore, a lot of recent research activities are aimed at multi-modal data management, trying to provide a complex yet computationally efficient representation of data objects by combining multiple views on object similarity.

In this thesis, we study the multi-modal retrieval in the domain of digital images and its utilization in the context of large-scale applications. We focus on the development of multi-modal search methods exploiting content-based retrieval and on evaluation of their applicability in two real-world scenarios – general web image search and automatic image annotation. In both of these scenarios, real-time processing is expected: therefore the search efficiency is one of the most important qualities. In addition to the novel retrieval methods, we propose a general query language for similarity-based multimedia retrieval. Furthermore, we provide a thorough performance analysis of different approaches to multi-modal image searching in

the context of large-scale retrieval. For the evaluation and deeper understanding of applicability of different solutions, a new evaluation platform for large-scale image searching was created. Finally, we present the MUFIN Annotation Tool software, which provides automatically generated keywords that describe image content.

Keywords

content-based image retrieval, similarity searching, multi-modal retrieval, information fusion, large-scale data management, multimedia query languages, retrieval evaluation, automatic image annotation

Contents

Chapter 1

Introduction

Information management and retrieval has always been one of the key challenges in computer science. In the early decades, researchers mainly focused on searching in text and simple attribute data, which represented the majority of existing digital information. In recent years, however, we have witnessed a massive increase in popularity of more complex data types and, in particular, multimedia. To mention but a few examples, there are probably more than 1 billion videos available on YouTube, with the upload rate of 48 hours per minute. Each day, about 250 million photos are uploaded on Facebook, and the Flickr web gallery hosts more than 6 billion images. The need for efficient means of data organization is greater than ever but so is the complexity of this task. Multimedia data are significantly different from textual and numerical in many aspects, making it necessary to develop novel tools for data management.

One of the key challenges in multimedia data processing arises from the complexity of the content, which cannot be indexed and searched as it is. Considering the video data example, we are not interested in pixel-to-pixel matches but rather in higher-level relationships, such as the similarity of visual content or semantic relations. This introduces a novel problem of finding a suitable level of content description that would balance the need for simplicity (from the computation costs perspective) and the need for complexity (for the sake of the best simulation of human understanding). Definition of a query is also not as easy as in text searching, where the query has the same format as the searched content and can be matched almost directly to it. In multimedia searching, queries are typically expressed by keywords or example objects and we need to bridge the gap between such query and the dataset. Furthermore, we need to face a number of more traditional information retrieval challenges such as creating suitable index structures, dealing with low quality of data sources, or meeting demands for interactive, easy-to-use retrieval engines, as well as personalization of the search process.

Apart from the problems and challenges, though, the modern information explosion also brings unprecedented possibilities. Enormous amounts of data coming from different sources in various formats, quality, level of detail, etc. allow us to obtain much more complex information about almost anything. All this data is generally available and waiting at our fingertips. On such premises, applications of artificial intelligence, such as automated recognition of objects, become a real possibility rather than fiction. Better access to information, then, allows the end users to obtain more advanced knowledge and subsequently to make better decisions. In this thesis, we hope to add a few tiles to the complex construction of information management systems of the modern era.

Large-Scale Image Search

Among all the complex data types that modern people produce, digital images are one of the most popular and frequently used. In numerous application domains, ranging from science to entertainment, we encounter digital photography, digital art, medical imaging, visual-based security applications, and many others. Due to the low costs and high availability of image capturing technologies, the bulk of existing digital image data and the speed of its growth are enormous. The need for efficient image data management is therefore indisputable. The ability to search image data and organize them with respect to their mutual visual and semantic similarity is essential for many applications that are needed today, such as automatic image classification and annotation, copyright infringement detection, filtering of inappropriate web content, landmark recognition, etc.

From the scientific point of view, the image retrieval problem represents a multi-faceted challenge as it stands at a crossroads of at least three disciplines – psychology, image processing, and data management. Even though images as a communication channel are natural to human cognition, we do not fully understand how people perceive visual information. Therefore, we first need to employ psychology to gain insight into the human understanding of images and their utilization in our thinking. Then we can switch to the image processing perspective and study ways of extracting the important characteristics from the image so that it can be used in further processing. Finally, we need efficient tools to handle the data and facilitate the searching.

In our research, we are mainly interested in the data management part of the whole process. Taking into account state-of-the-art accomplishments

in image understanding and preprocessing, we study the data structures and algorithms suitable for efficient retrieval. We also try to identify the fundamental patterns in the retrieval process and propose adaptable search methods that can be easily adjusted for different image descriptors or even other multimedia data, such as video or sound.

The data management issues are especially important in applications that need to deal with large quantities of data in real time, therefore we focus on these. In smaller or batch-processing tasks, the major challenge rests in obtaining high-quality results, which can be achieved by means of machine learning and fine-tuning the visual descriptors. For large-scale general-purpose retrieval, entirely different approach is necessary. Apart from the volumes, we typically need to deal with the diversity of data (both in content and quality) and with far more fuzzy specification of user needs and expectations. Therefore, we have to create flexible solutions rather than fine-tuned ones, which implies that a lot of query processing needs to be evaluated in real time. In such situation, the role of data structures and retrieval algorithms is crucial.

Thesis Structure

The thesis is organized as follows.

- In Chapter 2, we provide an overview of the current open problems in image retrieval, formulate the specific research objectives of the thesis, and briefly discuss the methodology of our research.
- At the beginning of Chapter 3, we clarify the concept of modalities in image searching and discuss possible approaches to the representation of images. In the following sections, we present three basic categories of modalities and review the principles of the related search paradigms.
- The survey of retrieval methods is continued in Chapter 4, which focuses on multi-modal image retrieval. We provide a comprehensive classification of existing approaches and detail the individual techniques.
- Chapter 5 presents our contributions to several issues related to multi-modal image retrieval. We propose, implement and evaluate three novel solutions for approximate multi-modal image search. Furthermore, we perform a unique, extensive evaluation of existing image

search solutions over large-scale real-world data. We also introduce a proposal for a multimedia query language.

- In Chapter 6, we focus on performance evaluations in large-scale image searching. We introduce a new evaluation platform we created, as well as the methodology applied for collecting the ground truth data.
- In Chapter 7, we move our attention to some important applications of the image retrieval methods. In particular, we focus on search-based image classification and annotation. We present a working implementation of the MUFIN Annotation Tool and discuss our participation in Image Annotation Task of the ImageCLEF contest. We also provide an experimental evaluation of the annotation functionality.
- Conclusions of the thesis and possible future directions are outlined in Chapter 8.

Chapter 2

Objectives

As we have anticipated in the Introduction, image searching is a wide research field with many unsolved problems. The majority of these challenges is still present in the more specific task of large-scale general-purpose retrieval. In this chapter, we provide a brief summarization of the main open issues in image retrieval and formulate our specific research objectives. Furthermore, we outline our approach to the selected problems and introduce the research methods that were used.

2.1 Challenges in Image Retrieval

Several recent surveys on multimedia and image retrieval [27, 55, 97, 146] provide us with a good overview of current open problems in this area. In Figure 2.1 we depict some of the main topics in a schema that shows the position of the individual challenges in the process of image retrieval. Those problems that are more closely related to our research are depicted in more detail.

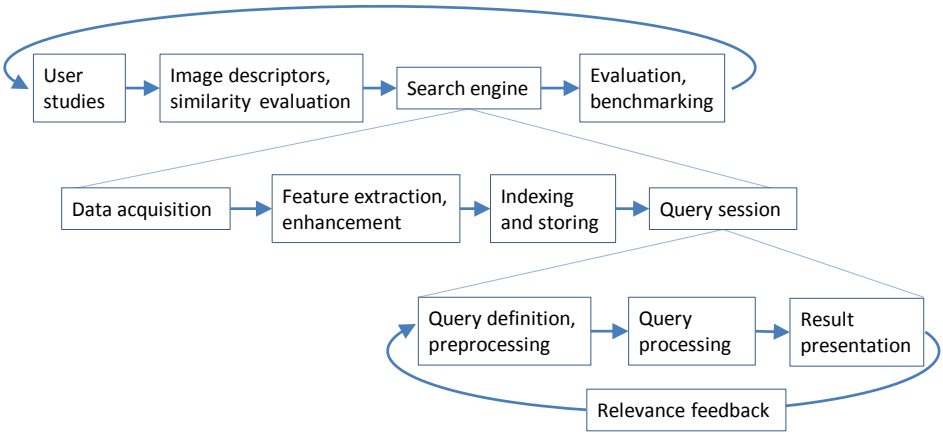

Figure 2.1: Selected open problems of image retrieval.

The tasks on the top level compose a life-cycle of a generic search system. Depending on the target application, users may have specific needs and expectations that need to be reflected by the choice of image descriptors and the respective similarity measure. Next, a search engine providing the desired functionality can be created. In the end, the performance of the whole system needs to be evaluated to identify directions for future improvements.

From the data management point of view, the most interesting part of the process is the architecture of the search system, which needs to deal with the topics of scalability, robustness, and performance. As the middle layer of the schema shows, we can identify several fundamental components of the search system. Three of them are involved in the acquisition and preparation of the dataset to be searched, whereas the last one represents the actual query session with user interaction.

In most situations, the two most important qualities of any search engine are the relevance of its results and the query response time. To achieve high results relevance, it is necessary to understand the user's information need, extract suitable and high-quality image descriptors, and perform as precise a search over the dataset as possible. At the same time, however, it is often necessary to guarantee low response times, which typically requires either massive computation power, or less detailed image descriptors and approximate searching. Finding a suitable trade-off between these two conflicting demands is one of the key challenges in image retrieval. Different query preprocessing and evaluation methods as well as the relevance feedback techniques are being developed to provide as precise results as possible within the imposed response time limits.

2.2 Focus of the Thesis

In the field of large-scale searching, high performance and scalability are of a crucial importance. To be able to meet these requirements, it is often necessary to accept some approximations in the processing of queries. However, this does not have to produce any noticeable negative effect on the quality of results because of two reasons: first, the understanding of general image similarity is always individual and no similarity measure can ever be precise in principle; second, really large datasets are likely to contain many relevant objects for a given query, therefore we can afford to miss some in the approximate searching. In other words, precision is much more important than recall in large-scale retrieval.

Accepting the need for approximations, there are two steps in the retrieval process where we can apply them – the choice of image descriptors and the query evaluation. Clearly, each descriptor is already an approximation of the original data object. Specialized small-scale applications can profit from complex descriptors where the approximation is not as significant. However, the complex descriptors typically require considerable storage space and take a lot of time to process, which is not feasible for interactive large-scale retrieval. Therefore, less specialized and smaller descriptors are usually employed to index and search large datasets. The second approximation takes place during query evaluation when some parts of the dataset may be skipped in the retrieval process even though they may contain relevant data.

Using approximations, we do trade search accuracy for performance. However, it is often possible to compensate for some of the precision loss by combining several different approximate searches. Moreover, the individual approximate searches can often be processed in parallel, thus keeping the overall response time low. In fact, a combination of different approximate approaches to searching can yield better results in large-scale retrieval than one fine-tuned search method, as the search space is too broad to fit into one universal descriptor and similarity measure. When a combination of several similarity measures is used, it is also possible to allow users to influence the final combination of the partial similarities, thus personalizing the search. Actually, flexibility of the search process is another highly desirable feature, especially in large-scale, broad-domain retrieval. For image search in particular, recent research studies [55, 89, 97] highlight the profits that can be gained by combining orthogonal approaches to image understanding, using the visual content, descriptive text, information about geographical position, and any other information available. The individual approaches are frequently denoted as modalities of image searching and the combined approach is known as multi-modal retrieval.

In this thesis, we focus on efficient ways of implementing the large-scale multi-modal search. Using state-of-the-art descriptors of image metadata and content, we study the possible means of query evaluation, which comprise various combinations of index structures, distributed query processing, and result postprocessing. This issue is vital for the whole task of image retrieval since the query evaluation may be used repeatedly in one user session, being part of many query expansion and relevance feedback strategies [86, 105, 120, 168].

The effectiveness and efficiency of any search method needs to be evaluated in the context of its target application. In our research, we consider two

possible utilizations of large-scale image retrieval – web image search and automatic image annotation. Both these applications respond to real-world user needs [94].

In the research presented in this thesis, we focus on the particular needs of image data management. However, we purposefully employ general data models in all the solutions we propose, so that these techniques are easily adaptable to other data domains. In particular, we utilize the concept of a metric space to model data, which guarantees a wide applicability of our solutions [165].

2.2.1 Contributions

The specific contributions of this work are the following:

New methods for efficient query processing We study and develop two different approaches to multi-modal image retrieval. First, we examine the behavior of the Threshold Algorithm in large-scale real-world image retrieval and propose an approximate implementation of this algorithm. Then, we focus on another approximate strategy that combines a content-based retrieval with postprocessing techniques that exploit complementary modalities. We propose several new ranking methods and evaluate their performance. Furthermore, we also propose a novel scalable solution for result postprocessing. All the new methods are added to the existing MESSIF library, which provides support for similarity searching over metric spaces.

Query language for similarity searching Even the most advanced search system may become useless if its users are not able to access most of the functionality due to a very complicated, low-level communication interface. Since the scope of available query types, search algorithms and various user settings is becoming more and more complex in contemporary multimedia search systems, simple visual interfaces are no longer sufficient for the definition of similarity queries. Therefore, we propose an extension of the SQL query language that adds support for similarity searching into the language.

Categorization and evaluation of query processing methods We survey existing approaches to multi-modal retrieval, identify significant design patterns, and introduce a systematic classification of techniques. State-of-the-art search methods as well as our own solutions are then analysed in

an extensive study with user participation. Insights into the real-world behavior of the search methods as well as guidelines for their application are derived from the experimental results. As a part of the study, we also create a new evaluation platform for web image searching which enables different research groups to share test data and cooperatively expand the ground truth.

Application of large-scale image retrieval in automatic image annotation We propose and implement a software tool that utilizes large-scale content-based image retrieval to select descriptive keywords for an arbitrary input image. The performance of this tool and the parameters that influence the precision of results are analyzed. We also discuss the usability of search-based annotation methods for classification problems and describe our participation in an image classification task of the ImageCLEF contest.

2.3 Methodology

Information retrieval in the domain of large-scale multimedia data is a very complex process with many aspects that influence the overall performance. Because of the numerous factors involved in the query processing, it becomes hardly feasible to model and analyze the retrieval performance in a strictly theoretical way. For efficiency, we can provide theoretical bounds on the minimum and maximum number of comparisons, disk accesses etc., but the real processing costs depend strongly on the properties of a particular data set (or perhaps the properties of a certain type of data sets – further research yet needs to be carried out in this area). As concerns the quality of search results, we can argue whether the results of a particular search methods will be precise with respect to a given similarity measure used within the search method. However, precision measured in this way may not correspond to user-perceived quality of results, as the human understanding of multimedia content is not yet fully understood. In the frequent case of approximate searching, there are usually no theoretical limits on the error of results. At best, we can derive the probability of error from the distribution of objects within the specific data set.

Therefore, we base our work on a combination of both the theoretical and experiment-driven research. We try to provide a complex view of the different approaches to multi-modal searching and analyze the advantages and disadvantages of their design theoretically. On this level, the analysis is very general and can be applied to many different types of data that sat-

isfy the basic conditions we specify in our model. Further on, we assess the performance of the individual methods on real-world image datasets in real-world scenarios. To evaluate efficiency, we apply standard performance measures, such as query response time or the number of distance computations. Regarding the effectiveness issues, we provide quality estimations based on the measurable object properties as well as a detailed analysis of results obtained in extensive experiments with user participation. Using the experimental results, we strive to establish the relationships between search costs and quality for individual query processing scenarios.

Chapter 3

Single Modality Image Search

This chapter introduces the basic approaches to image retrieval, as they appeared chronologically. We first explain the fundamental concepts of image data management and identify three types of modalities. We do not go into much detail on those specific features used in image searching which are not of a particular relevance to our study, but focus on data management techniques used with the individual modality types. Each class of techniques is briefly introduced and we discuss its usability in different image search scenarios. At the end of this chapter, we provide a summary of strong and weak aspects of single-modality image retrieval.

3.1 Modalities in Image Retrieval

The concept of modality, although frequently used in research papers, is rather commonly understood than strictly defined. For the purpose of this thesis, we believe it is necessary to clarify the meaning of the basic concepts used in our study of image retrieval techniques. Therefore, let us begin with a short discussion of the terminology we use.

In the context of multimedia retrieval, the term *modality* is commonly used to refer to the format of a data transfer, or in other words to the "channel or system of communication, information, or entertainment" (Merriam-Webster dictionary). In the usual interpretation, the basic modalities in multimedia retrieval are image, sound, text, etc. In some application areas, however, a finer differentiation of modalities may be needed. In medicine imaging, for instance, modality denotes any of the various types of equipment or probes used to acquire images of the body, such as radiography, ultrasound and magnetic resonance imaging. Overall, *modality* refers to both the technology used to record the real-world object, and the semantics of the resulting data object. In the information retrieval community, the term *multi-modal retrieval* usually denotes a combination of several of the basic modalities, e.g. in text- and visual-based search.

Another term related to multimedia processing is a *feature*. Features represent various facets of objects that can be used for their management and retrieval. In case of images, the typical features are colors or shapes, but we can as well consider a GPS location, an image size, costs, popularity, etc. Features can be understood as functions that transform a complex object content into a less complex representation that is useful for data management purposes. Sometimes, the term *descriptor* is also used to describe the same concept [151]. In this work, however, we prefer to use the word *descriptor* to denote the physical representation of a feature, e.g. in a form of a vector of a given size and semantics. Features and descriptors are very frequently discussed in context of image processing and the phrase *multi-feature* searching is likely to be used for retrieval with respect to multiple visual descriptors of an image.

In this thesis, we use the established terms of single-modality and multi-modal retrieval. These notions have historical roots, relating to the fact that in the field of information retrieval, text was for a long time the only modality as well as feature. The concept of multi-modal retrieval can be traced back to the first attempts to involve more facets of complex data into the search process, typically text and image, text and video, etc. For such applications, most of the modality fusion approaches were intended. However, the same reasoning and techniques can be applied for the fusion of features relating to the same modality, e.g. color and shape visual descriptors. Therefore, we do not limit our solutions to the basic modalities as discussed earlier but understand the term *modality* in a broader sense in this work:

Definition. *A modality is any feature or a combination of features of a complex object that is perceived as an atomic information unit in the context of a given application.*

Now, a modality is defined by purpose rather than by data origin, which allows us to model retrieval systems in a more generic way. In accordance with the original meaning of the term, modalities remain basic building blocks of data management that can be used to construct more complex tools.

The following sections introduce three basic types of modalities, which differ significantly regarding the ways in which the corresponding descriptors can be indexed and searched. Solutions that only exploit one modality will be denoted as *single-modality* or *mono-modal* retrieval. *Multi-modal* searching will be discussed in the following chapter.

3.2 Attribute-Based Retrieval

A simple attribute-based representation of real-world objects and exact-match retrieval was historically the first approach used for data management. Even though the exact-match paradigm is not reasonably applicable to the actual multimedia objects, attributes can be well exploited for distinguishing between categories of complex objects. In the early days of multimedia data retrieval, attributes describing basic low-level properties of the data objects were mostly employed, such as image size, date of publication, or file type. More recently, attribute-style data is also being used to store selected information about object semantics, e.g. the number and location of faces in the image.

3.2.1 Principles

The term *attribute* traditionally denotes a simple descriptive feature of an object, such as the name or date of birth in case of a database of people [12]. Although this notion is also being used with a broader meaning in some contexts, we shall employ it to refer to these simple features. The value of an attribute is typically expressed as a single atomic value, mostly a number or a string. Attribute-based searching supposes that queries are formulated in a restrictive way and return objects that satisfy a set of conditions defined by user. These conditions are mostly related to some equality or ordering of values in the data domain of the respective attribute. We shall denote this type of queries as *relational database queries*. Noticeably, each database object is either strictly relevant or strictly irrelevant in this query paradigm.

3.2.2 Techniques

The values of attribute modalities can be produced manually or automatically. In the latter case, the function that transforms the complex object into such attribute value is often denoted as *classifier*.

Queries over attribute data can be evaluated very efficiently, using mature technologies of relational databases [144]. These techniques capitalize on the properties of simple domains, in particular the total ordering of values, which is exploited in index structures such as B+ trees.

3.2.3 Applicability for Image Retrieval

In image retrieval, two types of attributes can be used: elementary descriptions of image format and content, and semantic attributes. Image size, file

type, resolution or basic colors of the image exemplify the first group. These features can help to filter the dataset but are mostly not satisfactory for expressing user's information need. Semantic attributes, on the other hand, aim at providing information about the concepts identified in the image – number of persons, presence of a specific object, etc. [68]. Such information is valuable for the retrieval but is difficult to extract.

As the simple descriptive attributes are not capable of capturing the complex information of images and semantical attributes are difficult to obtain, attribute-based searching is rarely used as a stand-alone method of image retrieval. However, attributes are often used in combination with other modality. The combination of size- or time-based image filtering with keyword searching is common in many commercial image search systems, such as Google Images[1] or Bing Images[2].

3.3 Text Retrieval

Text-based searching in multimedia data has been studied since 1970 and was very popular in the beginnings of image retrieval [1, 141]. Mature text-retrieval technologies and user-friendly keyword-based query formulation are the strong aspects of this solution. On the other hand, the text-based searching can only be applied for a limited scope of multimedia data sources and applications, where the text metadata is available and carries enough information to enable the retrieval.

3.3.1 Principles

As its name suggests, text-based searching exploits the text metadata associated with a multimedia object. As opposed to the attribute-based approach, the text data is not required to follow any fixed structure. Typical sources of text data associated with an image object are the image title, annotation, tags, text on the surrounding web page, URL, etc.

Queries in text-based systems typically consist of several keywords or a short text, which describes the user's information need. Again, the queries do not follow any fixed structure. The objective of the system is to provide objects most similar to the query and there is no longer a strict distinction between relevant and irrelevant objects. Results are typically ranked by the likelihood of relevance. Query reformulation and expansion techniques are typically used to compensate for the complexity of natural languages.

[1] http://images.google.com
[2] http://www.bing.com/images/

3.3.2 Techniques

Similar to attributes, the text metadata can also be obtained in two ways, either manually or automatically. In case of automatic annotation, techniques based on machine learning are mostly applied, even though automatic image annotation based on similarity retrieval can also be considered (this approach will be discussed in more detail in Chapter 7).

Prior to the retrieval itself, both the source data and the query need to be normalized, i.e. transformed into a form suitable for searching. This comprises a number of techniques, including data cleaning, stemming or lemmatization, which transform a document into a set of *index terms* [12]. A lot of work has also been devoted to query disambiguation and enrichment in order to determine the semantics of the query and reflect it in the search process. The success of these methods is strongly influenced by the application domain (narrow domains are easier to work with) and the quality of available metadata (more information can be mined from long and high-quality text than from erroneous web tags).

The basic text retrieval models, as classified in [12], are the following: Boolean model, vector model, and probabilistic model. In the Boolean approach, documents and queries are represented as sets of index terms and the retrieval strategy is based on Boolean queries and binary relevance decisions. While simple and clear, this model is not suitable for most real-world queries. In the vector model, the data is represented by vectors with non-binary weights of the individual index terms and the resulting set of documents is ranked in decreasing order of similarity. The similarity of two documents is typically computed as the *cosine of the angle* between the respective vectors. *Term frequency* and *inverse document frequency* measures determine the weights of the index terms. The probabilistic model is based on the theory of probability and assumes iterative query evaluation, with the user providing feedback which enables to improve the answer, eventually reaching the ideal answer set. Even though the performance comparisons between vector and probabilistic model do not provide clear indications of supremacy of any of these, the vector model is the most popular, especially in the web search context. *Inverted files* represent the most common index structure.

Even though the basic techniques of text search are long-established, text understanding is still in the center of lively research. One of the popular directions concerns utilization of ontologies in the retrieval process. An *ontology* defines a set of representational primitives which model a domain of knowledge or discourse [103]. The representational primitives are typ-

ically classes, attributes, and relationships. Ontologies are not necessarily related to text data but are mostly used with this approach. Ontology-based retrieval tries to determine some relevant classes for a given document and exploit the relations to obtain semantical information, which is further employed in the search process.

3.3.3 Applicability for Image Retrieval

The applicability of text-based retrieval techniques to image data management is naturally limited by the availability of high-quality text descriptions. These are difficult to obtain, especially with large-scale datasets. On the other hand, commercial web search applications (Google Images etc.) prove that it is possible to obtain significant amounts of relevant text data via crowdsourcing, i.e. exploiting information provided by billions of internet users. However, a lot of issues remain to be solved. As debated in [94], one of the key tasks is to determine the relevant words in the surrounding text of the web image. These may also be biased by a specific purpose of the creator. Furthermore, the text processing research still tackles the problems of understanding the semantics expressed in a natural language.

Nowadays, text-based searching is probably the most popular approach to image retrieval due to the easy and intuitive query formulation, which comes natural to internet users well familiar with text document retrieval. However, a well-known saying claims that "A picture is worth a thousand words" and it is hardly possible to provide an exhaustive text annotation. Furthermore, immense amounts of image data are not accompanied by any text data at all as manual annotations are too costly to produce and automatic annotations tend to work only in limited data domains. Such data is thus not findable by any text-based solution [94, 146, 97].

3.4 Content-Based Retrieval

Because of the problems related to text-based searching of multimedia objects, an orthogonal approach of content-based data management began to attract attention in 1990s [146]. Its main objective is to allow searching in any data, exploiting the actual object content with no need of additional metadata. Inspired by cognitive psychology, the content-based retrieval is based on the evaluation of similarity between data objects, which is inherent to human recognition and learning [164]. The last two decades witnessed intensive research in this field, concerning the extraction of infor-

mation from the multimedia objects as well as the development of tools for efficient similarity-based searching.

3.4.1 Principles

The content-based searching comprises a whole class of approaches that exploit various characteristics of complex objects to evaluate their similarity. Depending on the data and target application, different features can be extracted from multimedia objects. Each feature needs to be accompanied by a specification of a similarity measure that is used to compare objects. The similarity is often expressed by the dual notion of *distance*, which describes the dissimilarity of objects. The distance-based approach to similarity will be used throughout this work.

The content-based retrieval mostly follows the *Query By Example* (*QBE*) paradigm, in which the information need is described by a *query object* and the task is to find objects similar to the query object. A distance function d needs to be available to measure the similarity. In mono-modal retrieval systems, the distance function is fixed, whereas in the multi-modal solutions users may define the similarity measure together with the query object. According to the type of the query, additional requirements may be posed on the answer set.

The basic similarity queries are the *Similarity Range Query* and the *Nearest Neighbor Query* [165]. The Similarity Range Query $\mathrm{R}(q, r, \mathcal{X})$ is defined by a query object q and a radius r, and retrieves all object from a datasource $\mathcal{X}$ within the distance r from q:

$$R(q, r, \mathcal{X}) = \{o \in \mathcal{X} : d(o, x) \leq r\}$$

The Range Query is a useful tool in applications where users understand the similarity measure and are able to specify the query radius (e.g. in spell-correction, where misspellings with Edit distance 1 can be automatically corrected). However, this is often not the case with multimedia retrieval, where the distance measures are much less intuitive. In such situations, the Nearest Neighbor Query is more suitable. In the elementary version, it retrieves such object from $\mathcal{X}$ that is the closest to the query object. In a generalized case of the *k Nearest Neighbors Query* (*kNN query*), a user-defined number k of most similar objects is retrieved:

$$kNN(q, \mathcal{X}) = \{\mathcal{R} \subseteq \mathcal{X}, |\mathcal{R}| = k \wedge \forall x \in \mathcal{R}, y \in \mathcal{X} \setminus \mathcal{R} : d(q, x) \leq d(q, y)\}$$

Due to being intuitive and analogous to text retrieval, the kNN query is probably the most popular type of similarity query. Accordingly, we mainly

focus on this query type in our research. However, a number of other query types can be used in specialized applications, such as *Reverse Nearest Neighbor Query*, *Similarity Joins*, *Multi-object Queries*, etc. [158, 165]. With the exception of similarity joins, the answer set is typically ranked by the distance of the result objects from the query object(s).

3.4.2 Techniques

As discussed e.g. in [146, 27, 151], the key components of an effective and efficient content-based retrieval system are 1) the selection of representative features and descriptors, 2) similarity measure definition, and 3) data indexing techniques. In this section, we briefly introduce state-of-the-art solutions to these issues, focusing mainly on image data management. The content-based searching deserves a more detailed introduction than the previous approaches as its performance is usually the bottleneck of complex multimedia search applications.

Descriptors

The requirements for a suitable object descriptor are nicely summarized in [153]: "Good descriptors should describe content with high variance and discriminance to be able to distinguish any type of media, taking into account extraction complexity, the sizes of the coded descriptions, the scalability and interoperability of the descriptors." Unfortunately, two competing requirements occur in this formulation – high discriminance, and acceptable extraction and search complexity.

In state-of-the-art image features and related descriptors, we distinguish two basic types – global features and local features. A global feature characterizes the whole object by a single descriptor with a fixed structure (often a vector of numbers). Image color histogram is a typical example of such feature, more advanced features that describe image color, edges and textures are provided in the MPEG-7 standard [116]. Local features, on the other hand, describe a flexible number of characteristic regions of the multimedia object. In the domain of images, the two most popular local features are SIFT [106] and SURF [22]. Both of these select the characteristic keypoints from local extremes in an image, which are identified by different image enhancement filters. A single image is then represented by a set of descriptors representing the individual keypoints.

The local image descriptors are more detailed and in general provide better discriminance. On the other hand, the extraction as well as storage

and retrieval is much more costly, as the descriptors often require more space than the original image objects. Therefore, global image descriptors are mostly used for pure content-based retrieval, especially in large collections [16, 135]. Local features are employed in specialized applications such as subimage searching [81], or in result re-ranking strategies, which will be discussed in the next chapter. A more thorough discussion of visual descriptors can be found in [146, 55, 58].

Similarity Measures

The function that measures the distance (dissimilarity) of two data objects is a vital part of the content-based retrieval. Together with the feature descriptor, the distance measure models user's understanding of similarity in a given situation. At the same time, the distance evaluation should allow efficient data management.

For vector descriptors, the most straightforward similarity measure is the Euclidean distance, eventually other L_p metrics from the set of Minkowski distances. More sophisticated measures take into consideration specific properties of individual dimensions of the vector (e.g. Quadratic Form Distance or some of the similarity measures associated with MPEG-7 descriptors). Specialized measures also exist for other descriptor types than vectors (texts, graphs, sets, etc. [165, 94, 55]). In particular, local image descriptors are mostly processed by a set-based approach, using so-called *bag of words* technique. The individual feature vectors are transformed into a limited set of *visual words* and submitted to a text-like retrieval, and geometrical verification is then applied on the candidates.

Sometimes, a single similarity measure may combine information from multiple descriptors, e.g. color and shape features of an image. The combination can be either fixed for a given system, or flexible. The flexible similarity measures require the underlying system to be multi-modal, i.e. allow to handle the individual features independently. Such cases will be discussed in Chapter 4. With a fixed similarity measure, the combination of features behaves as a single modality. For instance, search system described in [16] employs a fixed combination of five MPEG-7 global features to describe the visual similarity of objects. Weighted sum is used as the aggregation function and the individual weights were determined by supervised machine learning [3].

Apart from being semantically relevant, the similarity measures also need to be suitable for retrieval. One of the important characteristics is the cost of a single distance computation, as distance evaluations are often

computationally intensive and may thus become a performance bottleneck of the whole search system. In addition, mathematical properties of the similarity measures may be very useful for data organization. In particular, a whole family of index structures is based on the properties of metric distance functions.

Indexing

Especially in large-scale applications, data indexing techniques are an essential component of the data management. In content-based retrieval, the efficient data organization is more challenging than with attribute- or text-based searching, which only operate on a limited set of domains with well-known properties. Content-based searching, on the other hand, should be applicable to any data or, at least, to a large scope of data domains and distance functions that are requested in real-world applications. The indexing techniques can be divided into several groups, depending on the restrictions laid on the descriptor domain and distance function properties [164].

Historically, the first approach to complex data indexing was based on the vector space model of the data. In this model, data objects are represented by points in a multi-dimensional vector space, which is then partitioned by various hierarchic structures, such as R-tree or k-d tree [141]. However, this approach is only applicable to vector descriptors and L_p distance functions, which are not satisfactory in some situations. A more general approach, based on the metric space model, can be applied on any data, provided the respective distance function satisfies the conditions of a metric. The properties of the metric space can be utilized for data space partitioning in several ways; e.g. the M-tree [47] is based on a pivot-based partitioning. A thorough survey of metric searching can be found in [165]. The metric space approach is satisfactory for a larger scope of problems but the constraints of distance symmetry and triangle inequality of metric spaces may still be too restrictive for modeling user-perceived similarity in some situations. A recent survey [145] discusses methods for efficient non-metric similarity search. An opposite research trend relies on transformations of complex objects into the domains of established index structures, which is applied e.g. in the previously mentioned concept of visual words that utilizes text-based search techniques.

As we already indicated in the previous section, the evaluation of distance between objects is often an expensive operation. In interactive large-scale data processing, it is often not feasible to evaluate a precise search and check the distances of all candidate objects. Therefore, approximate strate-

gies are frequently applied. These are mostly based on early termination, search space pruning, or probabilistic approaches. The space pruning can be exemplified e.g. by the M-index structure [124], which performs heuristic ordering of data regions by their expected relevance and then surveys them in this order, until a given limit of objects is visited. Probabilistic approach is also exploited in the Locality Sensitive Hashing (LSH) [88, 5] class of approaches, which gained a lot of attention several years ago. The LSH-based solutions rely on a sophisticated hashing function that guarantees that near objects are hashed to the same bucket with a high probability.

Similar to other data domains, the performance of content-based searching can be boosted by the utilization of distributed parallel processing. The discussion of distributed structures for metric-based searching is contained in [19].

3.4.3 Applicability for Image Retrieval

The generic design of content-based retrieval methods offers promising functionality for the multimedia retrieval. Since the descriptors and distance measures can be defined as needed, this solution is in principle applicable to any situation. However, the content-based approach is also accompanied by significant drawbacks.

The two major problems related to search effectiveness are denoted as *sensory gap* and *semantic gap* (a more detailed study of "gaps" in content-based retrieval can be found in [59]). The sensory gap refers to the difference between a real-world object and the information contained in the recording of the scene [146, 74]. We can illustrate this problem by considering two identical 2-D images of a 3-D object – there is no way of saying whether the two images depict the same object. The semantic gap, on the other hand, refers to 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." [146]. In other words, the semantic gap expresses the difference between the automatically extracted image descriptors and human understanding of multimedia content, which is based on a real-world experience and is not fully understood so far even by psychologists. Users can easily distinguish between an image of a yellow ball on grass and another image showing a dandelion flower, but it is not so easy to do so using general color and shape descriptors. Machine learning techniques can be applied to tune the similarity evaluations for a specific situation, but the resulting distance function will probably require

costly processing and will be one-purpose only, providing no means for accommodating possible subjective preferences of individual users.

Content-based retrieval also faces significant performance problems in the context of large-scale data management. A phenomenon denoted as a *curse of dimensionality* occurs with both vector space and metric space data organization. In a high-dimensional vector space, the partitioning often results in very sparsely populated data sections and the need to visit a large proportion of the whole dataset in each search. The retrieval performance then degrades to the efficiency of a sequential scan [94]. In metric spaces, the problem does not arise from vector dimensions but from the *intrinsic dimensionality* of the space [45]. In many real-world applications, the data distribution in the metric space is such that the distance between most object pairs is practically the same, which makes it difficult to define effective partitioning. Therefore, the only existing content-based solutions that aim at large-scale retrieval utilize approximate search strategies and work with global descriptors, which require less space and processing time than local descriptors.

Even though similarity-based associations are natural to human thinking, we are not yet used to similarity-based information retrieval. Thanks to ubiquitous text retrieval, users are familiar with keyword queries, which are formulated on the conceptual level and express semantics of the required objects. In content-based retrieval, it is necessary to provide the query object, which is not as straightforward as typing the query text. The majority of commercial multimedia search applications therefore still relies on text-based searching as the basic retrieval technology. Content-based searching as a standalone solution is only used in prototype, mostly academic applications [156]. Other reasons for the low popularity of content-based retrieval, such as the high complexity and costs of the creation of such system, are discussed in [164].

3.5 Summary

In this chapter, we have briefly surveyed the foundations of three types of single-modality retrieval: attribute-based, text-based, and content-based. Table 3.1 highlights and compares the most significant features of these paradigms.

The attribute-based approach only allows category browsing and is thus suitable only for small-scale applications or as a complementary feature. Most of the existing image search applications exploit text-based or content-

	Attribute-based retrieval	**Text-based retrieval**	**Content-based retrieval**
Search paradigm	Relational database queries	Similarity searching	
Applicability	Metadata needed (error-prone, often not available)		Always applicable
Supported queries	Category search	Keyword search (often expresses semantics well)	Query by example (semantic gap problem)
Performance	Efficient search processing		Data management may be costly in both time and space (depends on descriptor and distance function)

Table 3.1: Single-modality retrieval – a comparison of significant features.

based retrieval. Text-based searching is user-friendly and efficient, but can only be utilized when text metadata of a satisfactory quality is available. Content-based searching is a strong tool inspired by cognitive processes, but its practical usability is limited by the complexity of query processing – only simple descriptors can be employed in interactive large-scale retrieval, whereas sophisticated descriptors and distance measures are needed to overcome the semantic gap. Thus, none of the presented techniques provides adequate tools for all common needs of multimedia data management, especially in case of large-scale retrieval over broad data domains [55, 94]. Latest research trends suggest that better results could be obtained by techniques that combine multiple modalities, which are studied in the following chapter.

Chapter 4

Multi-Modal Image Search

In this chapter, we are going to study modern trends of multimedia data management, which have emerged from intensive research activities of recent years. Commercial subjects as well as academic researchers are busily developing search techniques, creating new tools, and organizing evaluation competitions. The number of existing solutions and the speed of their growth is so high that it is not possible to survey them all here. However, one phenomenon is common to many contributors: the effort to provide scalable, flexible and effective retrieval tools that integrate several complementary modalities of data representation.

We begin our discussion by an overview of arguments that lead contemporary scientists to believe multi-modal data management can bring significant improvements to complex data processing. Next, we formally define the multi-modal retrieval problem and introduce a notation system. Subsequently, a multi-faceted categorization of approaches to large-scale multi-modal searching is presented, which identifies and describes the basic types of modality-fusion solutions. The following sections provide detailed insights into the principles of individual approaches and a survey of related work. The concluding section then highlights some of the open challenges in multi-modal image retrieval which inspired our own research.

In accordance with our research interests (as stated earlier), we focus mainly on the techniques that are related to image retrieval, and devote special attention to the scalability aspects of the individual approaches. In the large-scale image retrieval, and the web image retrieval in particular, the two most frequent modalities are text descriptions and visual descriptors, which will be mostly discussed. However, it is important to realize that the same principles of orthogonal modality combination are also relevant for other complex data types and their respective modalities. Therefore, we also study solutions proposed for video or music retrieval. Unless stated otherwise, we assume the distance-based search model.

4.1 On the Importance of Being Multi-Modal

In the previous chapter, we have demonstrated that in spite of years of intensive research, approaches based on a single modality are not satisfactory for general multimedia retrieval. Some solutions are limited by the availability of metadata, other ones are too costly for practical use in a large-scale environment. However, a well-designed combination of several modalities can help to reduce the negative aspects and achieve the desirable qualities, i.e. availability, efficiency, and flexibility. Before we discuss such combined solutions in more detail, let us introduce the concept of the multi-modal searching in the following informal definition.

Definition. *Multi-modal data retrieval exploits multiple data modalities to describe complex data objects and evaluate their similarity. The individual modalities are expressed as independent descriptors and can be employed in different phases of the data management process. The procedure of combining partial object representations into a more complex evaluation model is often denoted as information fusion.*

Multi-modal data representation is a recent approach to the fundamental problem of multimedia information retrieval – i.e. finding such a representation of a complex object that carries all necessary information and at the same time is suitable for efficient data organization. Complex data objects cannot be sufficiently represented by simple modalities, such as text descriptions or basic visual features, as these do not carry enough information to satisfy user's requirements and are not distinctive enough to separate relevant objects from irrelevant ones. The multi-modal approach offers two solutions to this problem. Using the synergy between multiple modalities, we are able to define more sophisticated descriptors that model the objects on a higher semantic level. This approach, which integrates the modalities prior to the actual query processing, is traditionally denoted as *early fusion*. Alternatively, it is possible to model the retrieval process as an aggregation of results of several search runs, each of which exploits a different modality. This solution is known as *late fusion*. In both these approaches, the utilization of multiple modalities provides a more complex view of a given object, which helps us to overcome the semantic gap problem. Moreover, multi-modal retrieval also provides opportunities for cross-correction of errors in individual information sources.

Obtaining richer information about data objects is not the only advantage gained by multi-modal data management. Late fusion in particular provides two additional opportunities for designing effective and efficient

large-scale retrieval systems. First of these concerns the optimization of query processing costs. In contrast to solutions employing a single modality, the late fusion retrieval paradigm assumes that individual modalities are processed independently (at least to a certain degree). Parallel processing of partial queries is a natural solution that may significantly reduce the overall response time, as the individual modalities are supposed to be relatively simple and easy to process. Other optimization techniques, such as successive filtering of objects, will be discussed in more detail in the following sections.

Another strong advantage obtained by the utilization of multi-modal searching with late fusion is the flexibility of the retrieval process. It is important to realize that in the case of a general-purpose retrieval in broad data domains, it is not only technically difficult to obtain a single optimal modality for a given data collection – it is not even theoretically possible. When working with complex objects, the users' information need is often rather vague at the beginning and changes during the search process as the users are influenced by the information they continuously gain. This subsequently affects the users' understanding of similarity and their search preferences. Moreover, these preferences also depend on the particular context of the search – the same user can initiate a content-based search with the same photo several times, but expect information about the depicted people at one time and information about the landmarks at another time. Naturally, the problem becomes even more pronounced when we consider different users. Altogether, the concept of similarity, which is exploited in multimedia searching, is by nature subjective and context-dependent.

To reflect this phenomenon in the information retrieval, it is necessary to provide flexible data management tools. The most straightforward way is to allow users to adjust the distance function that evaluates the dissimilarity of objects. This requires the dataset to be indexed in such a manner that it allows efficient processing of queries with different distance functions. The multi-modal approach can support this requirement nicely by providing several data representations and similarity measures that are managed independently and can be (up to a certain degree) freely combined into a complex evaluation of similarity.

4.1.1 Modalities in Real World

The multi-modal representation is very well suited for many existing data sources, which often provide information in several modalities. This can be easily illustrated on video data, which consists of visual component, audio,

and possibly subtitles. However, multiple modalities can be found even for data where it is less obvious, as the digital information is becoming more and more connected. Apart from the content of the particular data object, the modalities can originate also from its context, usage, etc.

For image data in particular, it is natural for people to consider pictures in context of the situation they depict. Therefore, users typically accompany their photos by annotations or tags that specify the event, activity or place that was captured. In other datasources, text may be the primary information channel which is accompanied by images or other multimedia objects. To mention but a few significant examples, let us consider Wikipedia, the modern encyclopedia which contains enormous amounts of multi-modal information, or numerous social networks, which allow users to share multimedia data and enrich it by tags. Naturally, different data sources provide different range of modalities and variable quality of the data. The multi-modal techniques endeavour to utilize as much information as possible.

In web image retrieval, the solutions that combine text modality with either global or local image descriptors are very common. The text and visual representations are not only frequently available, but have been shown to be complementary to each other and to provide meaningful results [70]. Naturally, this solution is only possible when the text metadata is available. Depending on the specific situation, other features such as location, time, popularity etc. can be also utilized to provide relevant context and help to distinguish relevant objects from irrelevant ones [89]. All these modalities represent various facets of complex real-world objects that are naturally perceived by human observers.

4.1.2 Problematic Issues

As formalized in [117], an optimal multi-modal system should maximize the degree of heterogeneity as well as the information gain of each modality. Obviously, selecting such suitable set of orthogonal modalities and balancing their contributions to the overall similarity evaluation is a very challenging task. With a wrong choice of these inputs, the multi-modal approach may even decrease the effectiveness of a search system as compared to a single-modality solution. The authors of [117] identify the following two aspects that require particular attention in order to avoid degradation of the retrieval quality:

- The relevance of all modalities to be exploited should be verified to prevent the introduction of noise into the system.

- The fusion schema should be able to assess trustworthiness of the modalities towards the query in order to allocate confidence in modalities that have high relevance in the context of the query.

At present, most of existing systems do not provide automatic support for achieving these two objectives. The relevance of modalities is mostly decided beforehand by an analysis of the given dataset, while the query-specific adjustments are either not supported at all or left to the responsibility of users. Even with these simple approaches, we can mostly observe that the search precision is improved when multiple modalities are exploited. Developing tools for automatic identification of suitable modalities for a particular use case remains an open research topic.

4.1.3 Image Retrieval With High-Level Semantics

In this work, we mainly focus on developing effective and efficient retrieval techniques for a query defined by a multi-modal example. Such type of a query is already used in contemporary multimedia management applications, e.g. some web search engines. In the long term perspective, the research community aims at providing the so-called *semantic multimedia retrieval*, which is understood as a "retrieval by abstract attribute, involving complex reasoning about the significance of the objects or scenes depicted" [63]. Currently, there exist several research directions that try to narrow down the semantic gap and get closer to this goal. The overview study [105] classifies them into the following five categories: 1) techniques that use ontologies to define high-level concepts, 2) techniques that employ machine learning methods to associate low-level features with query concepts; 3) relevance feedback techniques that are utilized to learn users' intention; 4) solutions that generate semantic templates to support high-level image retrieval, and 5) approaches exploiting complementary modalities. All these methods need to be understood not as competing ones, but rather as complementary approaches which should be used in a cooperative manner in future advanced search tools.

Accordingly, we view the multi-modal retrieval techniques discussed in this work not as a universal solution to all multimedia search tasks but as a solution suitable for specific situations, which should be combined with complementary approaches when possible. It is out of the scope of this study to detail all the specific applications and suitable combinations of techniques. Therefore, we limit ourselves to mentioning the possible ties at the appropriate junction points.

4.2 Formal Model of Multi-Modal Retrieval

Before we start analyzing approaches to modality fusion, let us introduce and formalize the basic concepts and processes that take part in a multi-modal retrieval. First, we formalize the model of mono-modal retrieval that was discussed in the previous chapter, and then we extend this model to multiple modalities.

4.2.1 Single Modality Retrieval Formalization

Let $\mathcal{X}$ be the database of objects to be searched, which are of the type $\mathcal{D}_\mathcal{X}$. A modality $\mathcal{M}$ is represented by an ordered pair $(p_\mathcal{M}, d_\mathcal{M})$ of a projection function $p_\mathcal{M} : X \to \mathcal{D}_\mathcal{M}$, $\mathcal{D}_\mathcal{M}$ being a domain of modality $\mathcal{M}$, and a distance function $d_\mathcal{M} : \mathcal{D}_\mathcal{M} \times \mathcal{D}_\mathcal{M} \to \mathbb{R}_0^+$. The projection function transforms an object $o \in \mathcal{X}$ into a feature descriptor $o.f_\mathcal{M} \in \mathcal{D}_\mathcal{M}$, while the function $d_\mathcal{M}$ evaluates the distance between two descriptors, i.e. the dissimilarity of two objects as seen in the view of modality $\mathcal{M}$. Noticeably, this definition of modality fits all types of modalities discussed in the previous chapter, as the exact match paradigm can be easily reformulated in the distance-based terminology.

A mono-modal search engine $SE_\mathcal{M}$ stores each object $o \in X$ as a pair $(o.f_\mathcal{M}, o)$. The object descriptor $o.f_\mathcal{M} = p_\mathcal{M}(o)$ is exploited for indexing and retrieval of the data object o. The search engine $SE_\mathcal{M}$ may employ one or several index structures $I^1_\mathcal{M}, \ldots, I^n_\mathcal{M}$ to organize the descriptors of objects from $\mathcal{X}$. A query over $SE_\mathcal{M}$ is defined by a query object $q_\mathcal{M}$, which needs to be from the domain $\mathcal{D}_\mathcal{M}$. Alternatively, a query can be defined by an object $q_\mathcal{X} \in \mathcal{D}_\mathcal{X}$, which is then transformed into $q_\mathcal{M}$ by the projection function $p_\mathcal{M}$. Let us suppose that the most frequent query type, the kNN query, is issued. Then,

$$kNN_\mathcal{M}(q_\mathcal{M}, \mathcal{X}) = \{\mathcal{R} \subseteq \mathcal{X}, |\mathcal{R}| = k \wedge \forall x \in \mathcal{R}, y \in \mathcal{X} \setminus \mathcal{R} : d_\mathcal{M}(q_\mathcal{M}, p_\mathcal{M}(x)) \leq d_\mathcal{M}(q_\mathcal{M}, p_\mathcal{M}(y))\}$$

4.2.2 Multi-Modal Retrieval Formalization

For multi-modal data management, multiple modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$ need to be available. Each modality $\mathcal{M}_i$ is represented by the standard pair $(p_{\mathcal{M}_i}, d_{\mathcal{M}_i})$. During data processing, several modalities can be combined to provide more complex representations of objects from $\mathcal{X}$ and to evaluate their similarity on a higher semantic level. Let $p_{\widehat{\mathcal{M}_{i_1}, \ldots, \mathcal{M}_{i_m}}}$ be a multi-modal

projection function that transforms an object $o \in \mathcal{X}$ into a complex descriptor $o.f_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}} \in \mathcal{D}_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$. Since more such functions can exist, we further introduce $\Pi_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ to represent the set of all possible projection functions that can be defined over modalities $\mathcal{M}_{i_1}, \ldots, \mathcal{M}_{i_m}$. In a similar manner, we define a multi-modal distance function $d_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ and the set $\Delta_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ of all possible distance functions that exploit the given set of modalities. Both $p_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ and $d_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ can be defined in various ways, taking into consideration selected mono-modal projection functions and distance functions of individual modalities, but also e.g. the properties of the dataset $\mathcal{X}$. A more precise definition of several frequently used projection and distance functions will be provided later.

Let $SE_{\mathcal{M}_1,\ldots,\mathcal{M}_n}$ be a multi-modal search engine that recognizes modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$. $SE_{\mathcal{M}_1,\ldots,\mathcal{M}_n}$ provides the following tools to manage data objects from $\mathcal{X}$: a set of multi-modal projection functions π, and a set of multi-modal distance functions δ. The set π contains all supported projection functions from $\Pi_{\widehat{\mathcal{M}'}}$ for all possible $\mathcal{M}' \subseteq \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$. In the same way, δ contains all supported distance functions from $\Delta_{\widehat{\mathcal{M}'}}$. The functions from π and δ are utilized for data management and retrieval. In further discussions, we shall use the symbol $I_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ to denoted an index structure that exploits modalities $\mathcal{M}_{i_1}, \ldots, \mathcal{M}_{i_m}$, and $CS_{\widehat{\mathcal{M}_{i_1},\ldots,\mathcal{M}_{i_m}}}$ to denote a set of candidate objects retrieved by similarity defined in fusion of $\mathcal{M}_{i_1}, \ldots, \mathcal{M}_{i_m}$.

A query $Q = (q, d_Q)$ over $SE_{\mathcal{M}_1,\ldots,\mathcal{M}_n}$ is defined by a query object q and a distance function d_Q. The query object q can be specified as $q_{\mathcal{X}} \in \mathcal{D}_{\mathcal{X}}$, by a single modality descriptor $q_{\mathcal{M}_i} \in \mathcal{D}_{\mathcal{M}_i}$, or as a combination of several modality descriptors $(q_{\mathcal{M}_{i1}}, \ldots, q_{\mathcal{M}_{im}})$. The query distance d_Q needs to be taken from the set δ of supported distance functions. Let us assume that the query object is issued as $q_{\mathcal{X}} \in \mathcal{D}_{\mathcal{X}}$ and the query distance function is of the type $d_Q : \mathcal{D}_{\mathcal{X}} \times \mathcal{D}_{\mathcal{X}} \rightarrow \mathbb{R}_0^+$. Then, the multi-modal kNN query is defined as follows:

$$kNN_{\mathcal{M}_1,\ldots,\mathcal{M}_n}((q_{\mathcal{X}}, d_Q), \mathcal{X}) = \{\mathcal{R} \subseteq \mathcal{X}, |\mathcal{R}| = k \wedge \forall x \in \mathcal{R}, y \in \mathcal{X} \setminus \mathcal{R} : d_Q(q_{\mathcal{X}}, x) \leq d_Q(q_{\mathcal{X}}, y)\}$$

4.3 Categorization of Approaches

This section is devoted to a systematic categorization of multi-modal search methods, which introduces main directions of large-scale multi-modal image retrieval. A more thorough analysis of particular retrieval methods will

then be provided in the next section. Some of the observations in the following text were inspired by discussions of fusion techniques in multimedia processing survey studies [11, 27, 57, 95] and also by several research works that deal with information fusion in different domains [25, 138, 139]. However, none of the existing works addresses the problem of modality fusion in context of large-scale retrieval, which has its specific challenges. Therefore, we introduce a scalability-oriented taxonomy of multi-modal retrieval methods, which constitutes one of the contributions of this thesis.

In this categorization, we do not aim to cover all aspects of modality fusion, which are numerous and many of them have been studied elsewhere. Instead, our taxonomy focuses on several dimensions of the fusion that we believe to be significant for large-scale retrieval. The dimensions are not orthogonal but rather interconnected, so that a single design decision often influences several of the properties we study. However, we prefer to analyze the individual aspects separately to see more clearly how the different types of solutions work and what are their strengths and weaknesses. The dimensions we selected are the following:

- *Integration of modalities*: In similarity searching, the general task is to produce a set of objects similar to a given query. To be able to do so, we need to define a composite similarity measure that integrates the individual modalities. The manner in which the modalities are composed to provide the final similarity evaluation determines the semantics of the fusion and is therefore one of the most important fusion characteristics.

- *Fusion scenarios*: The timing and implementation of the fusion is the most obvious characteristics in which individual fusion solutions differ. Moreover, this aspect is tightly related to the efficiency and effectiveness of the retrieval, which are the vital characteristics of any search system. There are two well-known basic fusion scenarios – the early and late fusion. We thoroughly study the principles, challenges and effects of both of these.

- *Flexibility*: The third dimension of our taxonomy studies flexibility of individual approaches. As we have discovered in the introductory discussion of fusion benefits, flexible search systems allow users to adjust the retrieval to their specific needs and are needed in many real-world applications.

- *Precision*: Relevance of results is the primary goal of searching. However, it is sometimes necessary to utilize approximate solutions to

satisfy efficiency requirements of large-scale interactive applications. The approximations can be applied on different levels, including the actual fusion procedure.

- *Efficiency and scalability*: To be able to process the enormous amounts of data that exist today (e.g. on the web), the retrieval techniques need to be efficient and scalable. Naturally, the retrieval costs are influenced by many factors. We identify the most significant ones and analyze the influence of the different fusion methods on the overall retrieval costs.

For each of these dimensions, we analyze the existing solutions, identify their important characteristics, and sort the approaches into subclasses where applicable. Even though we believe our selection contains the most important characteristics of multi-modal solutions for large-scale retrieval, it is by no means complete. Therefore, several additional aspects are outlined and briefly discussed in the end of this section.

4.3.1 Integration of Modalities

In the multi-modal search paradigm, several complementary modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are exploited to describe complex data objects and evaluate their similarity. During data processing and query evaluation, these modalities are combined together to produce the overall similarity measure d_Q requested for the particular query Q. The fusion process may take into account the individual data descriptors $f_{\mathcal{M}_1}, \ldots, f_{\mathcal{M}_n}$, the respective distance functions $d_{\mathcal{M}_1}, \ldots, d_{\mathcal{M}_n}$, or both. For different types of data, modalities, and use cases, different strategies to the composition of partial similarities may be suitable. In this section, we focus on the semantics of the individual solutions and outline some related research topics.

We find it convenient to divide the approaches to the integration of modalities into two groups: in the first case, all modalities are considered with approximately the same level of importance and processed in parallel until the moment of fusion, whereas in the second approach, some of the modalities are treated as more influential and are used to index the dataset and pre-select the candidate objects before other modalities are engaged. The choice between these two options, and the subsequent selection of integration parameters, depends on various properties of the input modalities (e.g. data quality, correlation between modalities, etc.) as well as the target application. Moreover, the design decisions concerning integration

may also be motivated by efficiency considerations, since the individual approaches can differ significantly in the processing costs.

Symmetric Combination

Under the approach we denote as a *symmetric combination*, individual modalities are processed independently up to a certain moment in the query processing, when all of them are put together. Even though the contribution of each modality can be increased or decreased by a particular setting of the fusion mechanism, all modalities are basically considered to be equally important and are used in all similarity evaluations.

The symmetric fusion may be realized by a combination of individual feature descriptors, by an aggregation of partial distance measures, or by a combination of both these methods. The following sections provide detailed descriptions of these techniques.

Feature fusion Feature (or descriptor) fusion is an integral part of early fusion strategies, which combine modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$ prior to data indexing. After some analysis of the mono-modal descriptors $f_{\mathcal{M}_1}, \ldots, f_{\mathcal{M}_n}$ of data objects in $\mathcal{X}$, a new complex projection function $p^{FF}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$ is defined together with a distance function $d^{FF}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$ that evaluates the overall similarity of objects. This projection function and distance measure then determine the semantics of the searching. Let us specify the type of the overall distance function here, as it helps to show the difference between the feature fusion and the other symmetric fusion solution that will be presented later:

$$d^{FF}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} : \mathcal{D}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} \times \mathcal{D}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} \rightarrow \mathbb{R}^+_0$$

For feature fusion, the main challenge is the maximal possible exploitation of the relations between modalities which should ensure that the resulting similarity measure is semantically relevant. A typical representant of recent research directions in this field is a latent semantic analysis of relationships between visual words and keywords, which will be discussed in more detail later.

Distance aggregation The term *distance aggregation* denotes the process of combining the distance measures $d_{\mathcal{M}_1}, \ldots, d_{\mathcal{M}_n}$ of individual modalities directly into the overall similarity measure $d^{AG}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$. A fused projection

function is not needed in this case. Distance aggregation takes place in some early fusion solutions (other utilize native distance functions of the domain of fused descriptors) and in most late fusion techniques. In case of the late fusion, some authors [139] write about *parallel mode of operation*, since the individual modalities are processed in parallel until the moment of fusion. The type of the aggregated distance is the following:

$$d^{AG}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} : (\mathbb{R}^+_0)^n \rightarrow \mathbb{R}^+_0$$

The most frequent aggregation type is a weighted sum of the partial distances induced by individual modalities, but many other aggregation functions have been proposed [11, 17, 57]. Parameters of aggregations, such as the weights of individual modalities, are mostly determined by dataset analysis and machine learning techniques [16, 17, 163]. Alternatively, users can personalize the search by setting the respective weights manually, if the system architecture supports flexible aggregations.

Distance normalization The aggregated distance $d^{AG}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$ between objects o_1, o_2 is computed over their partial similarities $d_{\mathcal{M}_1}(p_{\mathcal{M}_1}(o_1), p_{\mathcal{M}_1}(o_2))$, $\ldots, d_{\mathcal{M}_n}(p_{\mathcal{M}_n}(o_1), p_{\mathcal{M}_n}(o_2))$. Balancing the contributions of individual modalities is typically requested before the actual aggregation, which is often a challenging task. The mono-modal similarity between the query object and any given dataset object can be expressed as the distance between these two items as computed from their descriptor values, or alternatively as the rank of the dataset object in a sorted list of results defined by the given modality. In the first case, we speak about *distance-* or *score-based aggregation*, whereas the second case is denoted as *rank-based aggregation*. The distance-based strategies are more common [57], but require the partial distances to be normalized, as the ranges of values of the respective distance functions may not be compatible.

The straightforward normalization by the maximum and minimum possible distance values is the most common, but it may not always be appropriate since is does not take into account the real distribution of distances in a dataset. Therefore, authors of [14] propose another solution, where the distance histograms of individual modalities are taken into consideration. Alternatively, machine learning techniques can be applied to choose the aggregation parameters, in which case the differences between distance ranges are inherently compensated by the learned weight settings.

Asymmetric Combination

Asymmetric modality combination strategies constitute a complement to the symmetric solutions. Here, the modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are not considered equal but instead, one or several of the modalities are chosen as dominating or *primary*. Let us suppose that the modalities are ordered in such a way that $\mathcal{M}_1^P, \ldots, \mathcal{M}_m^P$ are the primary ones. These modalities are applied in data indexing phase to organize the stored data, and in a search session to pre-select a set of candidate objects $CS_{\mathcal{M}_1^P,\ldots,\mathcal{M}_m^P}$. This candidate set is then subjected to further evaluation, where *secondary* modalities $\mathcal{M}_{m+1}^S, \ldots, \mathcal{M}_n^S$ as well as the primary ones may be exploited. Noticeably, such solution typically results in an approximate retrieval, as the query result $\mathcal{R}$ is often evaluated in the following way:

$$CS_{\mathcal{M}_1^P,\ldots,\mathcal{M}_m^P} = k'NN_{\mathcal{M}_1^P,\ldots,\mathcal{M}_m^P}(Q', \mathcal{X})$$
$$\mathcal{R} = kNN_{\mathcal{M}_1,\ldots,\mathcal{M}_n}(Q, CS_{\mathcal{M}_1^P,\ldots,\mathcal{M}_m^P})$$

Here, k' denotes the size of the candidate set $CS_{\mathcal{M}_1^P,\ldots,\mathcal{M}_m^P}$ and Q' the query object transformed into domains of values of the primary modalities. For obvious reasons, this approach is sometimes also denoted as incremental *filtering* of the dataset, study [139] refers to it as *serial mode of operation*. Clearly, a result $\mathcal{R}$ obtained in this way may not exactly satisfy the definition of a kNN query as defined in Section 4.2. In particular, some false dismissals of relevant objects may occur in the first phase of retrieval, i.e. during the selection of the candidate set. We shall discuss this phenomenon in more detail later. We can also notice that with the asymmetric fusion the issues of finding a suitable aggregation function and normalizing the partial distances may not be relevant. In case of two modalities, the primary one may be used solely to select the candidates, which will then be searched using the secondary modality only. Naturally, it is also possible to combine both modalities in the second phase, in which case the same problems need to be solved as discussed in the previous section.

Reasons for applying the asymmetric fusion may be threefold: the primary modalities may really be more vital for a given use case scenario, the asymmetric solution may be chosen because of efficiency issues, or some of the modalities may not be available at the beginning of the query evaluation. Let us explore each of these situations more thoroughly.

The situation in which some of the modalities are more important can be easily illustrated in the following use case of a restaurant search: Paul wants to find a restaurant that suits his preferences and is located within a

walking distance from his hotel. Evidently, it is not necessary to compare all restaurants in the world to his preference list; instead, only those that satisfy the locality criterion should be taken into consideration. Therefore, locality becomes the primary modality, whereas the similarity between a restaurant's description and Paul's dining preferences will be the secondary retrieval criterion.

Asymmetric fusion techniques are also frequently used to provide a simple and efficient multi-modal solution. In this case, the modality with the highest selectivity or fastest processing is chosen as the primary one. Most typically, a text modality is used in this place since text retrieval tools are well established. Only the objects with relevant text metadata are then further evaluated by more costly modalities that e.g. take the content into consideration. From the efficiency point of view, the advantages of this approach are indisputable. However, it is necessary to carefully analyze the input dataset and target use cases to decide whether the filtering of objects by the primary modality will not have significant negative effects on the recall of the system. This may easily happen if the primary modality is of a low quality – it is well known that text-based search of web images will miss a large portion of relevant objects, as these are not associated with the appropriate keywords.

The situation in which some of the modalities become available only in the course of query evaluation is the most interesting. The values of such modality are defined in the context of a given query, using pieces of information obtained in the query processing. To illustrate this concept, let us consider a music search scenario: Jane wants to find songs similar to her favourite one, so she submits this song into a content-based audio search system. The system supports several audio descriptors, but also categorizes songs by genre. As Jane has not defined her favourite genre, this information cannot be utilized in the beginning of the search. However, after the identification of candidate objects, the system can analyze them, identify the genre most probable to suit Jane, and filter the candidate songs.

The last scenario introduces a new dimension to our discussions of multi-modal retrieval methods. We need to distinguish between modalities that are available from the beginning of the retrieval process, and the additional modalities that arise during the query processing. Therefore, we introduce the concept of basic and derived modalities:

- *Basic modality* $\mathcal{M}^B$ is any modality that is readily available in the dataset and is independent of a particular query. Typically, the basic modalities are used to index the dataset.

- *Derived modality* $\mathcal{M}^D$ is a type of information that is not explicitly contained in either the query or the dataset, but can be derived during the query processing from the properties of candidate results.

4.3.2 Fusion Scenarios

For each of the basic approaches to modality integration we have identified, different implementations exist that vary in a number of aspects. In this section, we are going to focus on the timing of the fusion phase, i.e. its integration in the data indexing and query evaluation processes. Different fusion scenarios that are discussed bellow significantly influence both the costs of the data retrieval and the adjustability of the search process.

Depending on when the fusion is executed, the multi-modal approaches are traditionally divided into two classes denoted as *early fusion* and *late fusion*. First of all, we are going to review the early fusion solutions, for which the most important processing takes place before data indexing. Then, we move on to techniques that exploit the modalities during the actual query evaluation. To understand the basic design patterns of individual solutions, we take a closer look at the query processing pipeline and describe its main components. Afterwards, we discuss fusion techniques that can be applied in individual query processing phases.

For completeness, let us mention that both early and late fusion techniques can be utilized in one search system. In that case, we speak about a *hybrid fusion* [11].

Early Fusion: Dataset Preparation and Indexing

The principal characteristics of early fusion methods is the fact that all available modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are combined prior to data indexing. During the whole existence of the search system, only one fused projection function $p_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$ and distance measure $d_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$ are utilized, which can be understood as a new fused modality. Early fusion is also denoted as *data fusion*, *feature fusion*, or a *joint features model*, because it happens on the feature level, before any decisions concerning the similarity of objects are taken.

The main benefit of early fusion paradigm is the offline processing, where extensive analysis of data properties may be evaluated. Having defined the new fused modality, the search engine is then built as single-modal, using standard indexing techniques and retrieval algorithms such as the ones reported in Chapter 3. Naturally, such index structures can be chosen that are optimal for the given complex descriptor and similarity measure.

Major disadvantage of early fusion solutions is the limited flexibility of the resulting search system. The combination of modalities is usually fixed in the index and cannot be adjusted to accommodate particular user's preferences. Even though some progress has been made towards providing index structures that support multiple distance functions [37, 46], the flexibility is still very limited. Thus, the early fusion solutions are more suitable for narrow domains and well-defined tasks than for broad-domains and general purpose searching.

Within the early fusion category, we can distinguish the following three types of approaches that differ substantially in the level of data preprocessing applied:

- No data analysis: The simplest possible fusion solution just concatenates the individual descriptors into one and uses this new feature to organize the data. The fused projection function is thus of the type $p_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} : \mathcal{D}_{\mathcal{X}} \rightarrow \mathcal{D}_{\mathcal{M}_1} \times \ldots \times \mathcal{D}_{\mathcal{M}_n}$. A simple distance function suitable for the new descriptor domain is used (typically, some L_p metric).

- Distance aggregation: More sophisticated fusion techniques also utilize feature concatenation, but they exploit machine learning techniques to find a distance function that suits the information needs of a given application. The overall distance measure $d^{AG}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}} : (\mathbb{R}^+_0)^n \rightarrow \mathbb{R}^+_0$ typically aggregates the distances provided by individual modalities, as discussed in Section 4.3.1.

- Feature fusion with semantic analysis: The most advanced early fusion strategies focus on mining semantic relationships between modalities, and identification of data characteristics that are most important with respect to a given data set and/or retrieval task. Typically, the resulting feature space has a lower number of dimensions than the input ones, so the early fusion can also be seen as dimensionality reduction technique. A suitable distance function also needs to be selected during the semantic analysis.

Query Life-Cycle

For approaches that do not exploit early fusion, all the important processing takes place during the query evaluation. This comprises not only the identification of candidate objects, but also query preprocessing or relevance feedback. Actually, the query processing has become a rather complex task,

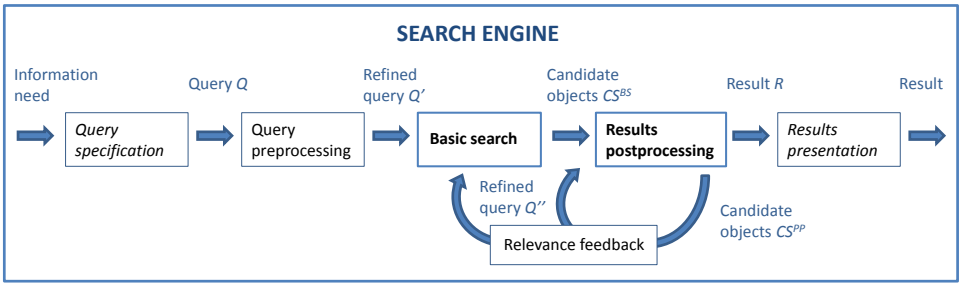

Figure 4.1: Query processing schema.

consisting of several distinctive phases that are depicted in Figure 4.1 and summarized in Table 4.1. The individual steps differ significantly in the amount and type of information exploited within them. Even though it is not necessary for each search engine to implement all these phases, the core structure of the processing remains the same across different solutions. For our further analysis of query processing techniques, it is essential to understand the fundamental mechanisms of the search task evaluation. Therefore, let us provide a brief introduction of each phase:

Query specification First of all, users need to express their information need as a query. Let us remember that a query is composed of a query object and a distance function, which together model the user's requirements. However, it is quite common that users only choose the query objects and the distance function is automatically provided by the system. The search engine interface should provide suitable support for intuitive query formulation, e.g. guide users through the definition of individual query modalities. No search engine computations are evaluated in this phase.

Query preprocessing Before initiating the retrieval process, it may be profitable to refine the query so that it is better suited for searching. In standard information retrieval, the query preprocessing typically comprises normalization of the search phrase, i.e. the removal of stopwords, lemmatization, etc. In the multimedia processing, an extraction of descriptors from query objects is a typical preprocessing activity. More sophisticated techniques can then be used to expand the query, i.e. provide richer information about the query object that can be exploited in the retrieval. A survey of query expansion techniques can be found in [39]. The costs of this phase may vary from negligible to considerable, if large knowledge sources are mined dur-

Query preprocessing	**Basic search**	**Results postprocessing**
Query is refined using additional information sources such as dictionaries, ontologies, or knowledge bases.	Dataset $\mathcal{X}$ is searched for candidate objects. Full scan or more sophisticated retrieval using index structures is employed.	Candidate objects are analyzed to reveal query and dataset properties. Additional information sources may again be used.
Input: query	**Input:** (expanded) query	**Input:** query and candidate objects
Output: refined query (cleaned, expanded, disambiguated, ...)	**Output:** candidate objects	**Output:** final result set (or a refined query in case of relevance feedback)
User interaction: possible	**User interaction:** none	**User interaction:** possible
Aim: provide as clear and rich query as possible	**Aim:** find candidate objects such that the most relevant are among them	**Aim:** identify the most relevant objects among the candidates
Costs: low to high, depending on the size of data analyzed and the complexity of processing	**Costs:** high for large data sources, depends on index structure used and cost of single object processing	**Costs:** should be low – similarity computations may be costly, but only a small dataset is processed
Critical issues: information sources need to be used carefully, otherwise the query may get puzzled rather than cleared	**Critical issues:** trade-of between results relevance and search costs	**Critical issues:** quality depends on relevance of input data; larger input better, but more costly to process

Table 4.1: Search phases comparison

ing the query refinement or expansion. In case the pool of potential query objects is known in advance, the preprocessing can be done offline for all objects. The phase produces a query object ready for searching.

Basic search The most critical phase from the efficiency point of view is the *primary* or *basic search (BS)*, when the candidate objects need to be identified from among the whole dataset. The costs of this phase depend on the size of the dataset, the index structures in use, and eventually on the applied level of approximation. The basic search is evaluated without user interaction and produces a set of *candidate objects* CS^{BS}. Depending on the strategy of the search engine, the candidates may be either directly forwarded to the user as the final result, or submitted to the postprocessing phase. In the latter case, the number of candidates is typically orders of magnitude larger that the requested result set size.

Result postprocessing When this phase is implemented, its task is to re-evaluate the distances between the query and the candidate objects, using additional measures of similarity. Typically, more complex computations are used in this step as the number of objects subjected to evaluation is much lower than in the basic search phase. Interaction with users can be exploited to obtain additional information about the selected candidates. Unless the relevance feedback mechanism is applied in the following step, the postprocessing phases produces the final answer set.

Result presentation In most systems, the results are displayed in a list or grid, ordered by their distance (dissimilarity) from the query object. In more advanced interfaces, result clustering or reordering may be applied to provide a more user-friendly presentation [7, 61].

Relevance feedback The relevance feedback is a mechanism that enables an iterative refinement of the result set. In its original form, as introduced in [140], relevance feedback assumes interactive searching, where users repeatedly provide their opinion on the relevance of current candidate objects. More recently, a variation denoted as *pseudo-relevance feedback* was introduced, which replaces the user opinion by an assumption that the candidate objects retrieved in last iteration are likely to be relevant and their properties can be utilized to learn about the properties of the desired answer. With both interactive and automatic evaluation of results, the relevance feedback loop may be repeated several times. In each iteration, either the query object or the query distance measure is updated. The refined query is then reintroduced either to the basic search, or the result postprocessing phase.

Late Fusion: Query Processing

In a multi-modal search system that exploits late fusion, modalities $\mathcal{M}_1, \ldots,$ $\mathcal{M}_n$ are not fused in advance, but at the query evaluation time. We can think of this approach as of an on-request fusion – a late fusion system typically supports mono-modal retrieval over some of the available modalities as well as a set of multi-modal distance functions from which the user can choose. In fact, flexibility of searching is one of the most important characteristics of late fusion solutions. Late fusion is also frequently denoted as *decision-level fusion* as the decisions – i.e. partial distances provided by mono-modal distance measures $d_{\mathcal{M}_1}, \ldots, d_{\mathcal{M}_n}$ in context of retrieval applications – enter the fusion phase instead of the descriptors.

Late fusion can be implemented in various ways, taking advantage of different information available in different query processing phases. Figure 4.2 populates the individual query evaluation phases with processes that may be utilized to exploit the multi-modal nature of the query. For easier orientation, let us suppose that we only need to combine two modalities. This model can be straightforwardly extended to multiple modalities. Let us briefly comment on the semantics and mechanisms of modality fusion applied in individual phases:

Query specification The processing begins by the specification of a query, which may be multi-modal, or consist of only one modality. When a multi-modal query is issued, user may directly define the modalities (e.g. choose preferred colors or shapes in case of image searching) or provide a complex query object (image), from which the modalities are automatically extracted.

Query preprocessing Query preprocessing can be used for two purposes in the context of multi-modal retrieval: first, it can attempt to refine or expand the input modalities, and second, additional modalities may be obtained. To refine the input, relationships between the modalities as well as some additional resources can be utilized. To illustrate such situation, let us consider a query defined by a keyword and a visual example, which asks for images of an apple fruit. By itself, the keyword "apple" is ambiguous, but the visual example together with some knowledge base (e.g. ImageNet [56]) can be exploited for disambiguation of the term. As for the acquisition of additional modalities, we can recall the example of music searching and automatic genre determination. In both cases, the preprocessing actually introduces an auxiliary query, which is evaluated either over the same dataset, in which case we speak about (pseudo-)relevance feedback, or over some external knowledge base. Query expansion that provides additional modality for image searching is reported e.g. in [43, 136, 149].

Basic search and postprocessing In these two principal retrieval phases the dataset objects are surveyed and the result set is actually formed. As we already know, in early fusion solutions the available modalities are integrated in advance and the whole query processing utilizes multi-modal descriptors and data structures. In late fusion systems, the data retrieval begins as mono-modal but during the processing, either the modalities are

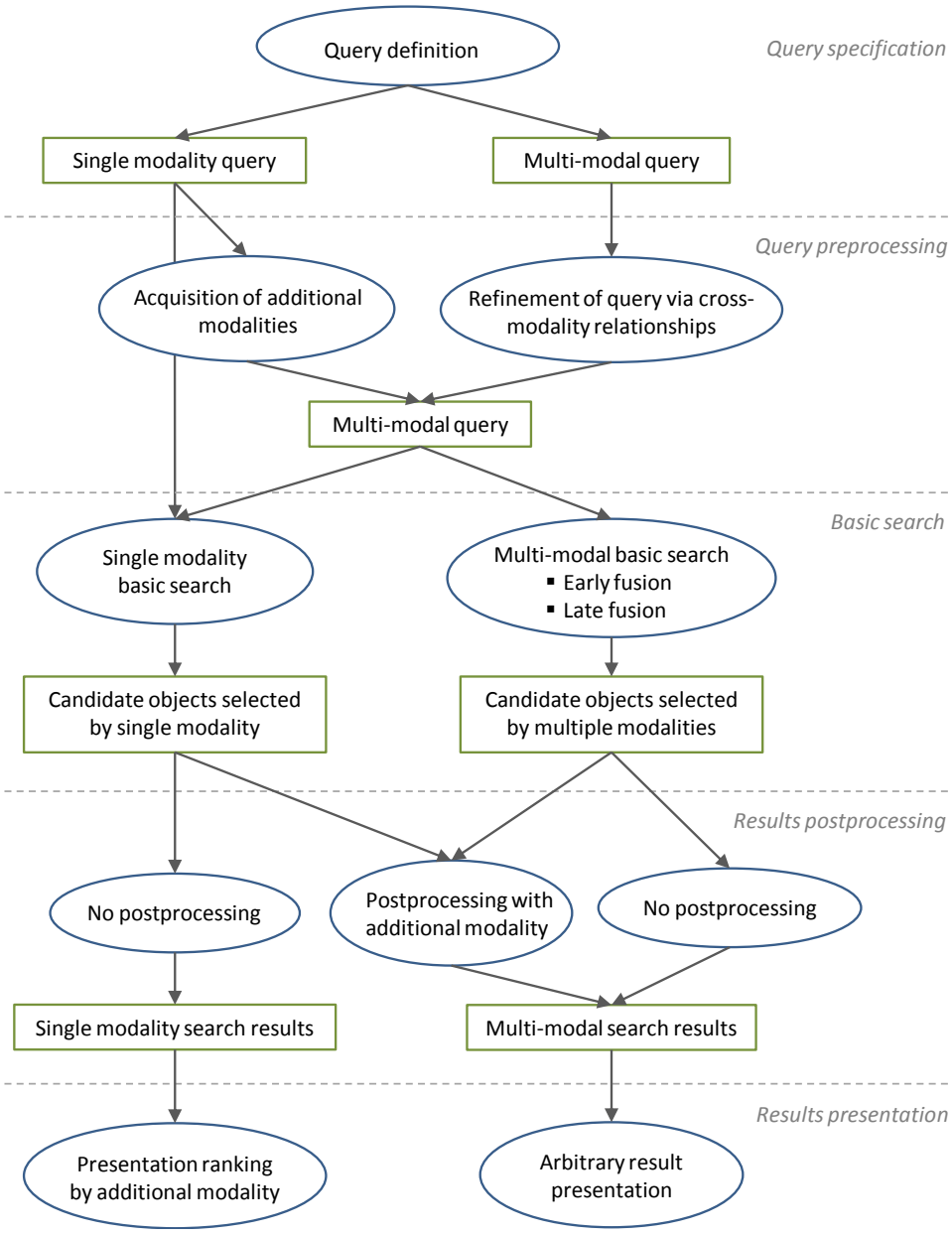

Figure 4.2: Modality fusion in the query life-cycle.

gradually added to the selection process, or several mono-modal candidate sets are being retrieved and merged. The actual fusion can occur in both basic search or postprocessing phase. The specific characteristics of each option will be further analyzed later.

Result presentation When all the evaluations are completed, the final result enters the presentation phase. As depicted in the processing schema in Figure 4.2, the whole query evaluation may exploit only a single modality until the presentation phase, then use the remaining modalities merely for the final organization of result objects. The result re-ranking applied here is very similar to some postprocessing strategies, but the lower-ranking objects are not discarded.

Relevance feedback To prevent the schema from becoming too complicated, Figure 4.2 does not consider the possible (pseudo-)relevance feedback loop in the query evaluation. When the relevance feedback is applied, all search iterations except for the last one do not produce a final result after the postprocessing phase, but return a refined query that re-enters the whole evaluation. Relevance feedback may be utilized to obtain values of some modalities that are not present in the query specification, to refine the values of available modalities, or to adjust the query distance function to better suit user's information need [98, 113, 143, 148].

Although most query evaluation phases may be involved in the integration of modalities, it is obvious that the basic search and postprocessing steps are the most important ones for the overall performance of a late-fusion system. These two phases form the retrieval core where candidate objects are retrieved and merged. Query preprocessing as well as relevance feedback build upon this core, issuing additional queries that are used to refine the query object q or to adjust the distance d_Q. The effectiveness of the whole searching is influenced by many factors, including the selection of modalities, aggregation function, and preprocessing and postprocessing strategies. Efficiency, however, is straightforwardly determined by the efficiency of the core phases. The more search iterations and additional queries are applied, the more important the efficiency of the core becomes. Therefore, let us explore the fusion capacities of these phases in more detail.

Late fusion in basic search phase In late fusion systems, the dataset $\mathcal{X}$ is typically preprocessed (indexed) by one or several of the supported modal-

ities. Let independent indices $I_{\mathcal{M}_{i1}}, \ldots, I_{\mathcal{M}_{im}}$ be available in a given system $SE_{\mathcal{M}_1,\ldots,\mathcal{M}_n}$. The basic search will typically begin by a retrieval of candidate objects from one or several of these indices. The fusion can be performed during this retrieval, or by merging the candidate objects from several separate mono-modal retrievals. In the former (asymmetric) case, the index structure $I_{\mathcal{M}_i}$ that is exploited needs to be designed in such a way that it can support retrieval by more modalities than were used for the organization of objects. The latter (symmetric) case is more general, as it allows to combine candidate sets $CS_{\mathcal{M}_{i1}}, \ldots, CS_{\mathcal{M}_{im}}$ from any number of independent searches. As the number of objects from the dataset $\mathcal{X}$ that may be accessed in the basic search phase is not limited, the processing may be quite costly. If sub-searches for individual modalities are used, these can be evaluated in parallel, but the actual fusion may become a performance bottleneck of the whole system.

Late fusion in postprocessing phase For fusion evaluated in the postprocessing phase, the basic search needs to provide a set of candidate objects CS^{BS}. One or several mono-modal basic search runs are therefore executed, each selecting a limited number of most promising objects from $\mathcal{X}$. These partial results are then merged together and form the candidate set CS^{BS}. No more objects are accessed in the postprocessing phase than those in CS^{BS}, which strictly bounds the processing costs. Depending on whether all modalities are exploited in the candidate selection, we speak about asymmetric or symmetric postprocessing fusion. The actual fusion is often denoted as *result ranking* or *re-ranking*, as the postprocessing defines a new ranking of the initial result, the top objects of which are then reported as the final result. It is important to notice that fusion in postprocessing phase allows to exploit derived modalities, as discussed in Section 4.3.1. Because of its low costs, ranking fusion is the most frequently used fusion technique in multi-modal retrieval [57].

4.3.3 Flexibility

At the beginning of this chapter, we have argued about the impossibility of defining a universal similarity measure for image data that would be suitable across different applications and user preferences. This poses a challenge of providing flexible search solutions, especially for broad-domain, general-purpose data management systems. Multi-modal approaches are in principle suitable for such needs, as they utilize several views on the

similarity. If users can influence the manner in which these are combined, their preferences get reflected in the retrieval process.

As we have observed in the descriptions of selected fusion scenarios, not all modality fusion techniques allow users to adjust the combination. Typically, early fusion approaches do not support flexible searching, whereas some late fusion architectures are highly adaptable. We propose to distinguish the following three levels of flexibility:

- *Zero flexibility*: The selection of modalities as well as their combination is fixed in the search system.
- *Aggregation flexibility*: The selection of modalities is fixed, but users can influence the aggregation function. The aggregation flexibility can be either *full*, or *partial*. In the latter case, the set of supported aggregation functions is determined by some required properties, e.g. monotonicity.
- *Feature flexibility*: In this case, we again distinguish between *full* and *partial* feature flexibility. For full flexibility, the modalities to be fused need not be determined strictly in advance, but users can choose an additional modality $\mathcal{M}' = (p_{\mathcal{M}'}, d_{\mathcal{M}'})$ during query specification. The system thus has to be able to introduce the new modality into a query processing without needing to rebuild the whole search infrastructure. In case of a partial feature flexibility, adding a new modality is processed off-line and may require adaptations of the infrastructure, but does not necessitate a complete rebuild of the search system.

4.3.4 Precision

The effectiveness of a retrieval system can be analyzed on two levels: 1) *distance-based* or *objective*, which analyses the precision of the query evaluation with respect to selected data representation and the query distance function d_Q, and 2) *user-perceived*, *subjective* or *semantic*, which concerns the satisfaction of users' information need. The second view is more realistic and determines the eligibility of the search system, but depends on multiple factors – the selection of modalities, quality of data capturing and feature extraction, the definition of the distance function, and the precision of the actual retrieval. In this section, we leave aside the semantical aspects and focus only on the objective precision of query evaluation.

In large-scale searching, it is very common that some approximations are applied in the query evaluation to decrease the computation costs. As

we discussed in Chapter 2, the approximations may not result in any noticeable deterioration of the result quality as perceived by users, because the similarity-based searching is always (semantically) approximate by nature. In the multi-modal retrieval, we need to distinguish between two types of approximations: those that regard the processing of individual modalities, and approximations of the actual fusion.

The approximations that can be applied during the processing of individual modalities, i.e. during the retrieval of candidate objects with respect to a given modality, are systematically studied and classified in a survey study [130]. The authors of this study analyze four dimensions of the approximate retrieval, which comprise the applicability a given technique for different data domains, the principles of achieving approximation, the result quality guarantees, and the user interaction with the system. The same criteria can also be applied on approximations that take place in the fusion phase. As concerns the applicability aspect, a wide range of solutions can be found; some are applicable to any data and distance function (typically, the solutions that fuse modalities in the postprocessing phase) while other are carefully tuned for a specific combination of features (some specialized index structures). If we focus on the implementation aspect, most of the fusion approximations fall into the category of *reducing comparisons*: the similarity of objects is not evaluated for all candidates that are potentially relevant, but only for the more probable ones. As for the results quality, usually no guarantees are given. On the other hand, the majority of fusion solutions allow users to influence the trade-off between retrieval costs and precision. More details about approximations will be discussed later for individual solutions.

4.3.5 Efficiency and Scalability

Efficiency and scalability are two interconnected topics that are obviously crucial for large-scale retrieval. Unfortunately, the efficiency of any search system is influenced by so many factors that it is nearly impossible to reliably assess the overall costs of the query processing. In this section, we try to decompose the processing costs into several parts and analyze the influence of different fusion techniques on the overall efficiency.

As we demonstrated in Section 4.3.2, query processing may consist of several phases, some of which can be evaluated more than once. Instead of analyzing the whole complex process, let us focus on the types of procedures that most significantly contribute to the overall costs.

Feature extraction The extraction procedure is typically applied in query preprocessing step. In this phase, it introduces fixed costs that do not influence system scalability. However, feature extraction can also be applied in later phases to obtain some additional features of selected candidate objects, in which case the efficiency of the extraction has a more pronounced influence on the overall performance. The extraction costs differ significantly for different modalities: while the text preprocessing is very cheap, the extraction costs of some content-based features such as the SIFT descriptors are considerable (more details can be found in Chapter 3).

Index traversal Selecting candidate object with the help of different data organization techniques (indices, hashing) is usually the most expensive task. Obviously, the costs are not static, but depend on data size, its distribution in the search space, and eventually on the approximate search strategy applied. Different indexing techniques as well as their limitations were discussed in Chapter 3.

Modality fusion So far, all the costs we discussed were related to the processing of individual modalities. However, the fusion procedure itself may also increase the complexity of query evaluation. Clearly, this only concerns the late fusion approaches. As we already mentioned, there is a significant difference between late fusion techniques that operate in the basic search phase and those that follow the postprocessing paradigm.

In the former case, it is very important how the results of individual sub-search runs are accessed and merged. As we shall see later, the number of objects that need to be visited in the fusion phase is not limited for some techniques and the fusion may thus degrade to linear complexity.

For solutions that exploit fusion in the postprocessing phase, the efficiency as well as scalability is much better. However, this is paid for by a lower search precision. The performance of individual postprocessing solutions may still differ significantly depending on the number of objects that enter the search phase, the necessity to extract additional features, and the complexity of similarity computations evaluated in this phase. However, the costs of the postprocessing remain fixed for any size of the searched dataset, and the scalability depends only on the performance of indexing structures exploited by primary modalities.

4.3.6 Other Aspects

The five criteria for classification of multi-modal retrieval techniques we have presented so far reflect our interest in efficient and flexible large-scale retrieval. Naturally, there exist other aspects that are worth attention and could be exploited to define a different categorization. Even though a detailed analysis of these issues is out of the scope of this work, let us briefly introduce at least some of them.

Selection of modalities As suggested in the brief discussion of potential dangers of multi-modal fusion in Section 4.1.2, it is extremely important to choose a suitable set of modalities that provides complementary information and does not worsen the retrieval effectiveness. From the efficiency point of view, it is also advisable to carefully balance the additional information provided by a complementary modality and the additional costs induced by its processing. Therefore, a thorough data analysis should be performed before a multi-modal search system is designed [11].

Suitability of individual fusion scenarios for different application domains This aspect is clearly related to the flexibility issue and also to the utilization of modalities during the fusion. Early fusion solutions that analyze data properties and exploit machine learning with ground truth data to select suitable data characteristics are very well fitted for narrow domains, but hardly applicable in general-purpose retrieval because of high processing costs. On the other hand, the late fusion approach that allows to construct flexible systems is suitable for broad domains.

Level of user participation in the retrieval process In our survey, we mostly focus on fully automatic solutions, as it is well known that in general, users do not like to provide much input during the query processing. However, in some specific cases the situation is different and the retrieval system can ask users for opinion on the relevance of selected objects. This information is then used to adjust the query processing, as demonstrated e.g. in [98].

Utilization of additional information sources Utilization of ontologies and general web data is a strong trend in modern multimedia retrieval. These resources find use mostly in query preprocessing. Solutions that exploit such data are reported e.g. in [80, 82, 120].

Synchronization of modalities In our study, we primarily focus on image data which do not have a time dimension. However, for stream data such as sound or video, the synchronization of modalities is a vital and very challenging task. A survey of related problems and available techniques can be found in [11].

4.4 Techniques

In the previous sections, we have analyzed different facets of multi-modal retrieval separately. In this part, we present several complete solutions and illustrate how the individual dimensions work together in practice. The following subsections represent various existing research directions in multimedia retrieval. For each of them, we provide a compact review of its behavior within the selected dimensions, present one or several particular techniques, and comment on the most interesting properties.

4.4.1 Simple Early Fusion

Fusion type:	*Symmetric aggregation*
Fusion scenario:	*Early fusion*
Flexibility:	*Zero*
Approximation level:	*None*
Fusion scalability:	*Medium*

The simplest possible solution for modality fusion is provided by a feature concatenation accompanied by some naïve similarity measure (e.g. an L_p metric) or a simple aggregation of the partial distances (e.g. a non-weighted sum or average). The concatenated descriptors and the aggregated distance form a single new modality that can be indexed and searched by standard index structures. Such solutions can be found among older solutions for the ImageCLEF retrieval tasks [57, 71] or video retrieval [147].

In these solutions, no semantical processing is performed to analyze the correlations between modalities – the modalities are considered orthogonal and equally important. The main advantages of this approach are its simplicity, easy implementation, and the fact that all modalities are exploited for all objects (this is a common trait of all early fusion techniques). Also, no training data is needed. However, many of the opportunities of multimodal searching are not exploited in the simple early fusion, as neither advanced semantics nor retrieval flexibility is provided. The efficiency and scalability of such solutions depends strongly on the performance of in-

dex structures that are utilized to organize the fused descriptors. However, the concatenation approach often results in bulky descriptors that face the dimensionality curse problem. Therefore, this type of fusion can be considered suitable only for situations where the modalities are really orthogonal, the descriptors are not too large, and user interaction with the search system is not expected.

4.4.2 Semantic Early Fusion

Fusion type:	*Symmetric, semantic fusion*
Fusion scenario:	*Early fusion*
Flexibility:	*Zero*
Approximation level:	*None*
Fusion scalability:	*Medium to high*

The main strength of the early fusion paradigm lies in the possibility of carrying out a thorough analysis of data properties. The synergy between modalities can be exploited to identify semantically relevant data characteristics and reflect these in the overall similarity measure. In this section, we present several techniques that perform such analysis and provide either semantically richer descriptors, or fine-tuned aggregation functions.

When exploited in early fusion, an aggregation function typically determines the semantics of retrieval over data represented by concatenated descriptors. Such solutions can be frequently seen in search systems that combine modalities of a similar type (i.e. multiple visual features) but are also used for a fusion of more diverse data (image visual content and text description). Feature concatenation with a weighted sum aggregation is used for the fusion of five visual descriptors in content-based search system MUFIN [16], where standard machine learning techniques were used to assess the aggregation function parameters. In [23], a similar approach is used to join text terms and visual terms for medical image search. The authors of [6] apply genetic programming methods to learn aggregation parameters for the fusion of local and global image descriptors.

Alternatively, the semantics can be contained in the descriptors. In this case, the fused descriptor is not formed by a concatenation of original descriptors; instead, significant dimensions from all modalities are identified and only these are contained in the new descriptor. To extract the significant characteristics, various statistical methods are used to analyze the joint feature space. Authors of [134] focus on latent semantic analysis, which is a technique frequently used to analyze documents in text retrieval, and ex-

tend it to work over multi-modal datasets. A similar approach is applied in [132], where text and visual modalities are fused by concatenating the columns of two unimodal matrices into a new matrix, which is then projected into the latent space to reduce the dimensionality. After projecting the query vector into the latent space, the cosine similarity is computed for each indexed document for ranking. A recent solution [65] fuses text descriptions of an image with labels that can be assigned to image regions by automatic annotation techniques. Distributional term representations are applied, in which terms are represented by a distribution of either co-occurrences over terms, or occurrences over other images.

Not surprisingly, the semantic fusion techniques outperform the simple ones in terms of result relevance. As for efficiency, the costs of data preprocessing are naturally higher, but that does not limit the scalability of the search system. When the semantic descriptor fusion is applied, the query processing may be even faster than in case of simple early fusion since the resulting descriptors contain less redundancy and are smaller than the concatenated ones. However, all the solutions reported in this section exploit fixed index structures and do not allow users to personalize searching.

4.4.3 Multi-Metric Indexing

Fusion type:	*Symmetric aggregation*
Fusion scenario:	*Early fusion*
Flexibility:	*Partial aggregation flexibility (monotone or linear agg. functions)*
Approximation level:	*None*
Fusion scalability:	*Medium*

The fact that users cannot influence the query processing is the main disadvantage of most early fusion techniques. Unfortunately, multi-modal indexing and flexibility are rather conflicting requirements. However, there are some fusion techniques that index data by multiple modalities and still allow for some level of flexibility.

An early proposal of such index system appeared already in 2000. The authors of [46] outlined the principles of an index structure called M^2-tree, which generalizes the data partitioning principles utilized in metric index M-tree to multiple modalities. The authors parallel the relationship between the M-tree and M^2-tree to that between the B-tree and the R-tree – the additional modalities work as additional partitioning dimensions in the M^2-tree. Multi-modal data indexed by the M^2-tree can be searched by any

distance aggregation function that satisfies the monotonicity property. The M^3-tree [38], on the other hand, uses a multi-modal distance function to organize the data. This indexing distance is designed in such a way that the resulting index can support queries over multiple query distances d_Q. In particular, only linear combinations of mono-modal distances are considered in this approach. The upper bound on the fused distance, i.e. an aggregation with all weights set to 1, is exploited in the indexing phase. To improve search efficiency (i.e. to achieve better pruning), the M^3-tree also stores the components of the indexing distance in the tree nodes, which allow computing tighter covering radii for subtrees during the query evaluation. The principal idea of the M^3-tree was further developed in [37] where the authors describe the mechanism of converting any metric index structure into a multi-metric one.

The presented indexing techniques provide a flexible, precise and relatively efficient retrieval solution. In particular, the efficiency of the M^3-tree is shown to be nearly equal to that of a standard M-tree built with the particular distance function requested by a given query. However, the efficiency may not be satisfactory for real-world applications as significant amounts of data need to be stored and complex distance computations must be evaluated during the retrieval. Moreover, these solutions are only applicable to metric data domains and selected types of aggregation functions.

4.4.4 Asymmetric Indexing

Fusion type:	*Asymmetric aggregation*
Fusion scenario:	*Late fusion, basic search phase*
Flexibility:	*Partial aggregation flexibility (monotone agg. functions), partial feature flexibility*
Approximation level:	*None*
Fusion scalability:	*Medium*

As we could see, all indexing techniques presented in the previous section were symmetric, i.e. all available modalities were used for both indexing and retrieval. However, it is also possible to find asymmetric basic search solutions that utilize a subset of available modalities to organize the dataset $\mathcal{X}$ but allow precise retrieval with respect to all modalities. This can be achieved by extending a standard mono-modal index with additional information about secondary modalities and adjusting the retrieval algorithm so that it takes these modalities into account when pruning the search space and identifying candidate objects.

To the best of our knowledge, solutions of this type have not yet been used in image retrieval but are studied in other domains, e.g. spatio-textual similarity search. The IR-tree [50] extends the standard R-tree spatial index to store both spatial and text information about points of interest. Non-leaf nodes of the IR-tree contain summarized information about text data in respective subtrees, which allows a search algorithm to prune the search space efficiently with respect to both textual and spatial modalities. Any monotone aggregation function can be then used to compute the query distance. In a similar way, the LBAK-tree [2] enriches the R*-tree spatial index. In this case, combined location- and approximate-keyword-based queries are targeted and the LBAK-tree consists of several types of nodes that maintain different types of information about text in respective subtrees.

Asymmetric basic fusion solutions show a potential for supporting precise multi-modal search with a level of flexibility sufficient for many applications, and relatively low additional costs in comparison to mono-modal indices. Noticeably, adding a modality to such index would not require rebuilding the whole structure – it would only be needed to enrich the nodes of an existing index tree with information about the values of additional modalities in the respective subtree. However, the solutions we have seen so far support only selected data modalities and have been proposed for particular applications with rather specific data distributions. Providing a general asymmetric basic-search solution and analyzing its applicability thus remains a challenging problem.

4.4.5 Threshold Algorithm

Fusion type:	*Symmetric aggregation*
Fusion scenario:	*Late fusion, basic search phase*
Flexibility:	*Partial aggregation flexibility (monotone agg. functions), partial feature flexibility*
Approximation level:	*None or guaranteed (possibly user-defined)*
Fusion scalability:	*Low*

The Threshold Algorithm is a well-known late fusion solution, which was introduced by Ronald Fagin in 2002 [66]. The aggregation of results takes place in the basic search phase, accessing as many objects as necessary to guarantee precise fusion results. To be applicable, the following conditions need to be satisfied:

- For each of input modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$, there exist two methods for accessing data objects. The *sorted access* is able to sort objects from $\mathcal{X}$

by their increasing distance $d_{\mathcal{M}_i}(p_{\mathcal{M}_i}(q), p_{\mathcal{M}_i}(o))$ from the query object with respect to the given modality, and report them one by one as requested. The object identifier and its distance are always provided. The *random access* is able to provide the mono-modal distance of any object when its identifier is issued.

- The aggregation function d^{AG} needs to be monotone.

Under these conditions, the Threshold Algorithm works as follows (the algorithm description is a near-exact quotation from [66] but we have transformed it to use our notation and distance-based terminology instead of original score-based):

1. Do sorted access in parallel to each of the n sorted lists L_i. As an object $o \in \mathcal{X}$ is seen under sorted access in some list, do random access to the other lists to find the distance $d_{\mathcal{M}_i}(p_{\mathcal{M}_i}(q), p_{\mathcal{M}_i}(o))$ of object o for every modality $\mathcal{M}_i$. Then compute the distance $d^{AG}(q, o) = d^{AG}(d_{\mathcal{M}_1}(p_{\mathcal{M}_1}(q), p_{\mathcal{M}_1}(o)), \ldots, d_{\mathcal{M}_n}(p_{\mathcal{M}_n}(q), p_{\mathcal{M}_n}(o)))$. If this distance is one of the k lowest we have seen, then remember object o and its distance $d^{AG}(q, o)$ (ties are broken arbitrarily, so that only k objects and their distances need to be remembered at any time).

2. For each list L_i, let $d_{L_i}^{max}$ be the distance of the last object seen under sorted access. Define the threshold value τ to be $d^{AG}(d_{L_1}^{max}, \ldots, d_{L_n}^{max})$. As soon as at least k objects have been seen whose distance is at most equal to τ, then halt.

3. Let $\mathcal{R} \subseteq \mathcal{X}$ contain the k objects that have been seen with the lowest distances. The output is the sorted set $\{(o, d^{AG}(q, o)) | o \in \mathcal{R}\}$.

The Threshold Algorithm represents a theoretically precise, clear solution that is applicable in many situations. It allows to combine results of independent search systems, which can be queried in parallel for the sorted lists. To add a new modality, it is only needed to provide a mono-modal search system that supports the two requested access operations. Noticeably, this mono-modal search can be provided by another party. Unfortunately, there are no reasonable limitations of the fusion processing costs. In the worst case, it is possible that the algorithm will need to visit all objects in the database to be sure that the optimal solution was found, which is not acceptable in large-scale applications.

To address this problem, Fagin also proposed an approximate variant of the algorithm. He defined a θ-approximation in the following way: For

$\theta > 1$, a θ-approximation to the top k answers is a collection of k objects such that for each $y \in \mathcal{X}$ among these k objects and each $z \in \mathcal{X}$ not among these k objects, $\theta.d^{AG}(q, y) \leq d^{AG}(q, z)$. Early termination algorithm as well as other solutions for situations with restricted sorted and random accesses are proposed in [66]. Unfortunately, there are still no guarantees of the retrieval efficiency. In the most scalable setting, a search system exploiting the Threshold Algorithm may iteratively display the changing answer set and the current precision guarantee and users can decide whether the searching (i.e., answer refining) process shall continue.

4.4.6 Symmetric Postprocessing

Fusion type:	*Symmetric aggregation*
Fusion scenario:	*Late fusion, postprocessing phase*
Flexibility:	*Full aggregation flexibility, partial feature flexibility*
Approximation level:	*Not guaranteed*
Fusion scalability:	*High*

Symmetric postprocessing fusion can be understood as an approximation of the Threshold Algorithm that accesses only a limited number of objects from each sorted list L_i provided by modality $\mathcal{M}_i$. The reasons for exploiting this approximation may be threefold: 1) the precise modality fusion evaluated by the Threshold Algorithm is too expensive, 2) the requested aggregation function is not monotonic, or 3) the mono-modal search systems that provide input for the fusion phase do not offer full sorted lists of objects from $\mathcal{X}$. We may also notice that solutions of this type are frequently used for small datasets where sequential processing of complete ordered lists L_i does not pose an efficiency challenge (e.g. [51]).

Symmetric postprocessing fusion is often employed in solutions of the ImageCLEF tasks [57], which typically exploit linear combinations of text and visual modalities. A more sophisticated solution of [40] utilizes fuzzy inference rules for score-based fusion. In [104], the CrowdReranking algorithm is presented, which combines results of multiple text-based web search engines to increase the relevance of text retrieval. The aggregation works on a voting principle known from classification algorithms. The authors of the CrowdReranking algorithm argue that although text search is used in all fused mono-modal systems, the particular distance measures are different and their combination provides richer information. A similar idea is discussed in [64] where a number of heterogeneous mono-modal methods are evaluated to maximize the diversity and complementariness

of searching. Multiple visual, textual and early-fusion multi-modal indices are exploited that use different information to search the data and produce the candidate sets. Finally, the authors of [96] propose to fuse the results of multiple retrievals over the same data, but different index structures. Each of these indices provides approximate results (because of efficiency issues), so several such results are combined to increase search precision. In this case, the partial results are aggregated by the union operator.

Easy applicability of the fusion phase on top of existing search systems and low additional costs are two obvious advantages of all postprocessing solutions. With a suitably chosen aggregation procedure, a symmetric solution can nicely compensate for possible bad performance of a single modality that can be caused e.g. by erroneous data. On the negative side, most postprocessing techniques produce approximate results with no quality guarantees. However, the approximation that is measured in terms of the query distance function may not have any noticeable negative influence on the quality of results as perceived by users. As we already know, a certain level of imprecision is inherently contained in the content-based data management and some false dismissals of relevant objects in query evaluation are acceptable, especially in large data collections. Finding the suitable sizes of the input candidate sets $CS^{BS}_{\mathcal{M}_1}, \ldots, CS^{BS}_{\mathcal{M}_n}$ that optimally balance retrieval precision and processing costs is a permanent challenge for each particular retrieval solution.

4.4.7 Asymmetric Postprocessing

Fusion type:	*Asymmetric aggregation*
Fusion scenario:	*Late fusion, postprocessing phase*
Flexibility:	*Full aggregation flexibility, full feature flexibility*
Approximation level:	*Not guaranteed*
Fusion scalability:	*High*

Asymmetric postprocessing fusion represents the approximate alternative to asymmetric solutions that operate in basic search phase. One or several primary modalities are exploited to provide the set of candidates $\mathcal{C}^{BS}$, which is then re-ranked with respect to additional (or all) modalities. Apart from the obvious option of re-ranking by modalities orthogonal to the primary ones, the asymmetric postprocessing also provides space for relevance feedback (RF) and pseudo-RF processing, which is frequently used.

Although some solutions that exploit visual features as the primary modality for image search are known [57, 98], the majority of re-ranking ap-

proaches are based on the well-established and efficient text retrieval. Such solutions are utilized e.g. in commercial image search systems Google [93] or Bing [160], which employ text retrieval to obtain the candidate objects and then reorder the results with respect to visual similarity. Interestingly, Google exploits only local visual features, while Bing prefers a combination of both local and global visual similarity. In both these systems, the re-ranking also takes into account the distribution of objects in the $\mathcal{C}^{BS}$, in particular their mutual distances, thus capitalizing also on the pseudo-RF information.

Ranking methods that exploit the pseudo-relevance feedback strategy are among the most rapidly developing solutions of the semantic gap problem. Numerous approaches try to extract some useful information from the initial result on the assumption that it should contain a substantial ratio of relevant objects. In general, there are two information sources contained in the initial result set: the properties of the candidate objects, and the mutual relationships between them. Both of these can be exploited during result postprocessing.

The object properties that are typically studied in the ranking phase are either their position in the search space (in case of the vector space model) or their distance from the query (the overall object distance as well as the partial distances for individual modalities). In both cases, the aim is to discover some important dimension or descriptor that shows low variance for many of the result set objects, indicating that this feature may be significant for the given query. With this knowledge, we can alter the similarity evaluation, making it more suitable for this specific query. This is a standard solution in classical relevance feedback [168] and is used for the ranking purposes in [41].

Exploration of relationships between the candidate objects provides another way of determining the relevant ones. Mostly, the assumption is that the relevant objects should be similar to each other while the less relevant ones will more probably be outliers in a similarity graph. Multiple approaches try to exploit this observation by the way of similarity graph processing. In [93, 137, 160, 167], a similarity graph is explored in a random walk. Solutions proposed in [83, 85, 113, 129, 169] apply various types of clustering, giving higher ranks to large clusters or to clusters which have their centroid near to the query object. A different approach is presented in [91], which proposes to use the idea of reverse-kNN queries and increase the rank of objects that have the query among their nearest neighbors. Alternatively, [131] studies the distances of candidate objects to other objects in the dataset and again looks for some significant patterns.

Similar to the symmetric postprocessing techniques, an important issue in re-ranking is the choice of the size of the input set $\mathcal{C}^{BS}$. In most solutions, this size is fixed, but the authors of [8, 9] argue that it is more advantageous to use a flexible size of the result set, deciding upon the distribution of item scores. Also, it is not clear whether the asymmetric postprocessing should consider both the primary and secondary modalities in the re-ranking phase, or exploit the primary modalities only for selecting the candidate set and then evaluate these with respect to secondary modalities only. The second option is more common, but [48, 98] demonstrate that the former approach may be more suitable in certain situations.

Re-ranking solutions are popular among contemporary multimedia retrieval systems as they can be implemented directly on top of an existing mono-modal retrieval system, e.g. a text-based search engine. The query processing can be very cheap if efficient index structures are available for the primary modalities. The RF and pseudo-RF ranking strategies also provide strong tools for overcoming the semantic gap problem. Pseudo-RF techniques have been more intensively researched recently because interactive relevance feedback, albeit very successful in increasing result quality, is not likely to be used by common users (a discussion of "lazy users" can be found in [79]). However, it is important to realize that the performance of asymmetric fusion strategies strongly depends on the quality of the candidate set provided by primary modalities. Therefore, the applicability of such solutions is limited to datasets where the primary modalities are available in sufficient quality.

4.5 Summary

At the beginning of this chapter, we have presented several theoretical observations that explain why a combination of modalities should be able to improve the quality of the complex data retrieval. In a decisive majority of research papers we have analyzed later, multi-modal solutions are shown to systematically outperform mono-modal ones in terms of retrieval effectiveness. Therefore, it is definitely worth pursuing this research direction. However, we could also see that there are many diverse solutions to modality fusion, which have specific properties, advantages, and disadvantages. Even though some approaches are known to be more suitable for certain types of applications than other ones, there is yet a lot of work to be done in understanding the fusion and deciding the most suitable solution for a given task.

Considering the particular subproblem of a general large-scale retrieval, represented e.g. by a web search, we know that the desirable properties are efficiency, flexibility, and relevance of top-ranking results, but it is not necessary to aim for a perfect recall. Therefore, late-fusion solutions are more likely to be chosen. This is confirmed e.g. by commercial image search engines that exploit approximate asymmetric late fusion. It has also been shown that users are often not willing to cooperate with the system and provide explicit relevance feedback. Subsequently, automatic selection of proper fusion models and further development of pseudo-RF methods are highly relevant research topics nowadays.

From a multitude of open challenges of the multi-modal image retrieval, let us highlight the following ones that particularly inspired our own research:

- Development of visual-based asymmetric fusion: The majority of early asymmetric solutions to multi-modal image retrieval were based on text as the primary modality. While effective and efficient, such approach is not applicable in situations when the text modality is not available or of a low quality. Using a content-based modality as the primary one poses bigger challenges in terms of efficiency, but may achieve better precision and recall of results. It is therefore desirable to develop postprocessing techniques for content-based basic search.

- Analysis of the trade-off between retrieval efficiency and the quality of search results: For many real-world large-scale applications, approximate searching is the only feasible solution since precise retrieval would require too much processing time. However, there are many parameters that adjust the approximate searching, the influence of which is not yet clear. Moreover, we should continue looking for novel index structures that would facilitate efficient data management with a limited level of imprecision.

- Analysis of applicability of different approaches to real-world tasks: The principal problem with application of multi-modal retrieval techniques to real tasks lies in our lack of knowledge about the suitability of these techniques for different situations. In particular, there are very few comparisons of the performance of different techniques on real-world data collections, which is caused also by the difficulties related to constructing a realistic benchmarking set. Furthermore, it is necessary to study the properties of different modalities as well as

datasets, and identify the characteristics that influence the usability of different search methods.

- User-friendly tools for multi-modal searching: As we could see, there are many shades of multi-modal searching and many parameters users may want to adjust to personalize the retrieval. A query language for similarity searching and other user-friendly tools need to be created to provide a unified way of communication with complex retrieval functionalities.

Chapter 5

Metric-Based Multi-Modal Image Search

In this chapter, we present our contributions to the problem of large-scale multi-modal image retrieval. These concern three different but interconnected areas: 1) research and development of novel efficient techniques of modality combination, 2) comprehensive evaluation of performance of different retrieval methods in real-world large-scale settings, and 3) a proposal of a query language for similarity-based searching with a support for multi-modal queries. As we could see in the summary of the previous survey chapter, all of these topics belong to highly actual open challenges of multimedia retrieval. Since we aim at providing flexible search solutions, we are mainly interested in late-fusion techniques in this work. We further focus on image data and strive to optimize the retrieval with respect to their characteristic properties. However, the solutions we develop are all based on the general metric space model and thus possibly transferable to other data domains.

As anticipated, the research in the area of complex data retrieval cannot be conducted entirely on a theoretical level, but must be complemented by an experimental verification of applicability of proposed solutions. Therefore, we begin this chapter with an introduction of the MUFIN retrieval system, a large-scale searching platform that we develop and employ to evaluate the usefulness of our techniques. In the following section, we present our contributions to the research of approximate multi-modal searching, in particular an approximate adaptation of the Threshold Algorithm, a set of novel postprocessing techniques, and an efficient solution for asymmetric fusion that scales gracefully with the underlying indexing structure. The third section is devoted to an extensive comparison of performance of different late-fusion solutions for image search, which was evaluated over real-world data with user-provided result relevance assessments. Finally, we introduce our proposal of a similarity search query language, which provides a general tool allowing users to issue any type of similarity query in a clear and unified way. We conclude the chapter with a brief summary of our achievements.

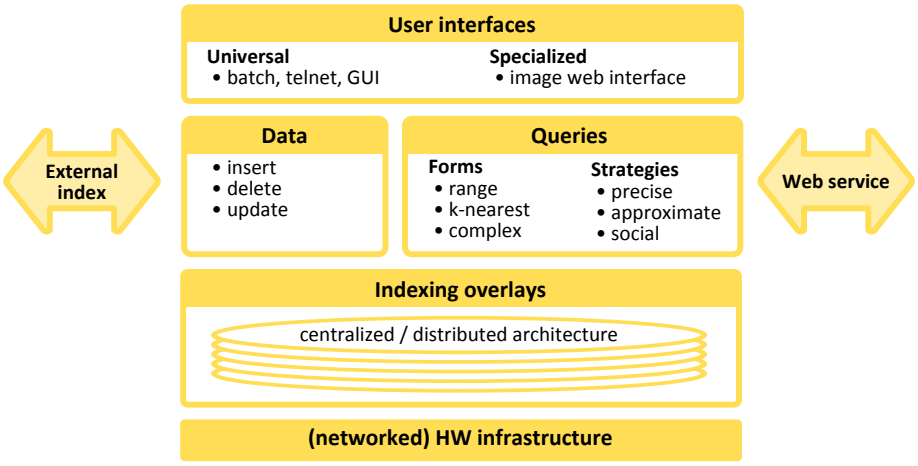

Figure 5.1: MUFIN four-tier architecture.

5.1 MUFIN Similarity Search System

MUFIN [15, 123] is a universal similarity search system, which is developed by our research team. Being based on a metric space model, the MUFIN system can facilitate retrieval over a wide range of data collections such as images, music, scientific experiments, or biometrics. One of the principal objectives of this system is to provide efficient large-scale retrieval. To achieve this, MUFIN supports both centralized and distributed data management and can evaluate approximate as well as precise queries. As a part of our research, we further develop the MUFIN search system, extending it with support for various types of multi-modal retrieval. Before we start discussing our contributions, let us briefly introduce the basic features of the MUFIN system and describe its particular instance – the MUFIN Image Search, which is utilized in all our experimental evaluations.

5.1.1 Architecture

The MUFIN acronym stands for *Multi-Feature Indexing Network*, describing the basic principles of the system design. The MUFIN system is able to index and search data with respect to multiple modalities (features), each of which defines an independent *overlay* of the system. These overlays form the executive level of a four-tier architecture depicted in Figure 5.1.

The lowest tier represents the hardware infrastructure the system is running on. Depending on the size of the indexed data and the efficiency re-

quirements, the hardware layer can be formed by a single machine, or a more complex infrastructure such as a cluster of dedicated machines, GRID or Cloud architecture, or a peer-to-peer network. The executive core of the MUFIN system in the second tier is formed by one or several indexing structures (overlays), which are mapped to the underlying hardware, potentially utilizing the distributed environment. Each of these overlays typically maintains only data necessary for a given modality (object keys and descriptor values), partitioned among its (logical or hardware) nodes. The number of logical nodes in respective overlays and their mapping to physical computers are the main parameters that affect the system's searching performance. From the third tier point of view, the logical nodes of all overlays form a single virtual overlay with a uniform access to individual members. The third tier provides interfaces for data maintenance (inserting and deleting data) and query specification, considering both the query form and the strategy for query execution. Finally, the fourth level comprises of a variety of user interfaces.

The MUFIN architecture also supports seamless integration of external index structures. Most typically, this functionality is exploited when text searching is requested by some MUFIN instance, since mature text retrieval techniques are readily available.

5.1.2 MESSIF Implementation Framework

The four tiers of the MUFIN architecture may be implemented in various ways, reflecting the needs of a given application. Individual search components are created within the *Metric Similarity Search Implementation Framework* (MESSIF) [21], a development platform for similarity searching created also by our research group. The MESSIF library provides basic operations for data management and enables easy implementation of queries. Moreover, it also enables sharing and reusing of the code as well as efficient testing and comparison of results. In particular, MESSIF provides the following functionality:

- encapsulation of the metric space concept – developers can use the data objects transparently regardless of the specific dataset, new data types can be added easily;
- concept of operations – introducing a uniform interface to manipulate and query the data;

- management of the metric data – storing objects in buckets with automatic evaluation of basic similarity queries and buckets-splitting based on the metric indexing principles;
- automatic performance measurement and collecting of various statistics, including a uniform interface for accessing and presenting the results;
- communication layer for distributed data structures – message navigation, automatic collecting and merging of distributed statistics;
- specialized load-balancing system for distributed index overlays;
- support for complex similarity queries in multi-metric spaces – this class of functions is further developed in our work;
- user interfaces – designed to control both the centralized and the distributed data structures and to present the retrieval results.

5.1.3 MUFIN Image Search

Since MUFIN is a general purpose software product, it can be applied to similarity search problems of a variety of applications. In order to organize a specific data collection, a MUFIN instance needs the following parameters to be specified for each modality $\mathcal{M}_i$: (1) the projection function $p_{\mathcal{M}_i}$ and the distance function $d_{\mathcal{M}_i}$ – since MUFIN is working with the metric space model, $(\mathcal{D}_{\mathcal{M}_i}, d_{\mathcal{M}_i})$ must form a metric space; (2) index structure for each of the modalities that are required to support self-contained retrieval – a local storage index or eventually a distributed indexing structure needs to be selected.

In this work, we develop the MUFIN Image Search system [16], a prototype application for broad-domain image retrieval. In the course of our research, we have worked with several modalities and indexing schemas. We briefly review them in following sections.

Modalities

In all our experiments, we limit our attention to two types of modalities that are most typical for image retrieval: the visual similarity of the image content and the textual similarity of image descriptions. For the visual similarity, we utilize global visual descriptors, as these are less costly in terms of both extraction and query processing and are therefore better suited for

MPEG-7 descriptor	Distance	Weight
Scalable Color	L_1 metric	2.5
Color Structure	L_1 metric	2.5
Color Layout	special	1.5
Edge Histogram	special	4.5
Homogeneous Texture	special	0.5

Table 5.1: Selected MPEG-7 descriptors, the respective distance measures and the weights for distance aggregation.

large-scale retrieval. In particular, we employ five descriptors defined by the MPEG-7 standard together with the distance functions recommended for them [116]. The specific selection of MPEG-7 descriptors, as shown in Table 5.1, is adopted from [67]. Although each of these descriptors can be used separately, they are typically combined to provide a more comprehensive evaluation of visual similarity. Authors of [67] propose to use a weighted sum combination and employ machine learning to determine the suitable weights. In our study [17], we also analyzed the relationships between these descriptors as well as several possible aggregation functions, including the simple weighted sum and a weighted sum of a logarithm of distances. Our experiments confirmed that both these combinations provide quite satisfactory results. In all experiments reported in this work, we utilize the simple weighted sum as specified in Table 5.1.

Concerning the textual modality, we experimented with two implementations. In the beginning, we compared sets of keywords related to images by the Jaccard set similarity measure [165], which expresses the ratio of matched keywords between the two images. This measure is a metric, therefore easily applicable within MUFIN, and can be evaluated very efficiently. However, solutions that employ this measure are not comparable with many other state-of-the-art search systems that exploit the standard cosine distance with *tf-idf* weighting [12]. Therefore, we adapted this measure in our later experiments.

Index Structures

In the course of our research, we studied the behavior of various retrieval techniques over different datasets. For some of these, centralized solutions were utilized, whereas other used distributed architectures to facilitate ef-

ficient retrieval over very large collections. However, the same indexing structures can be used in both situations, since the data partitioning principles are very similar. Currently, we utilize the Metric Index (M-Index) [124] structure, which is based on a hierarchical Voronoi-like partitioning of the search space. During the query evaluation, potentially relevant data partitions are accessed in the order of their probability of relevance. The M-Index can facilitate both approximate and precise retrieval – for approximate searching, the evaluation can be terminated earlier by specifying a fixed limit of the number of objects that can be accessed during a single query processing. In the case of distributed searching, the M-Index is applied on two levels – first to distribute the data among nodes of the underlying infrastructure, and then to organize the data for each node. More details about the distributed Metric Index can be found in [125]. Unless stated otherwise, the M-Index structure is utilized in all experiments reported bellow.

In some of our evaluations, we also utilize mono-modal text-based retrieval. As anticipated, textual search provided by external resources can be easily integrated into MUFIN. In our experiments, we employ the Lucene engine [111], a state-of-the-art open-source text search system.

5.2 New Solutions for Multi-Modal Retrieval

In our study, we focus on the large-scale, interactive image retrieval. In this context, one of the most important qualities of the retrieval process is its speed and scalability. At the same time, it is desirable to support flexible multi-modal searching as the current results indicate that this is a promising way to effective management of large data. Apart from being scalable, any such method needs to produce relevant search results. However, it is not required in the large-scale search scenarios that all qualifying objects are retrieved, which provides an opportunity for approximations.

In this section, we present our research in the field of late-fusion multi-modal retrieval, which comprises the following three techniques of approximate search: 1) an approximate symmetric fusion solution based on an adapted Threshold Algorithm, 2) asymmetric postprocessing fusion methods for image search that utilize visual image content as the primary modality, and 3) a novel inherent fusion technique, which provides an effective and scalable solution for asymmetric fusion in general. For each of these retrieval methods, we present a theoretical analysis of its benefits, provide a working implementation available within the MESSIF library, and experimentally verify its performance over real-world data.

5.2.1 Distributed Threshold Algorithm for MUFIN

As discussed in Section 5.1.1, the MUFIN system is able to index data objects with respect to multiple modalities (features) that can be accessed independently, thus allowing flexible searching with late modality fusion. A straightforward fusion solution for such situation is the Threshold Algorithm (TA), introduced in Section 4.4.5. However, the costs of the basic TA are too high to keep this algorithm applicable to large-scale interactive retrieval. Therefore, our objective is to provide an efficient search solution with symmetric late fusion for our system. In the following text, we introduce a distributed variant of the Threshold Algorithm that satisfies these requirements.

Multi-Layer Distributed System Architecture

To manage voluminous data collections, distributed data structures are supported within MUFIN. One of possible implementations of such structures is a peer-to-peer (P2P) data network, which is considered in this work. The peers (computers participating in the network) offer the same functionality and the system follows a shared distributed logic that facilitates an effective intra-system navigation. In general, every peer of such system must provide its storage and computational resources, must be able to contact any other peer directly (provided its network identification is known), and must maintain an internal structure that ensures correct routing among the peers. For maximal scalability, there are three fundamental requirements: data expands to new peers gracefully; there is no central node to be accessed when searching for objects; and the data maintenance primitives never require immediate propagation of updates to significant number of peers.

In the P2P architecture, every peer holds a partition of the indexed data collection in its storage area. This data can be stored either as a simple list, in which case a sequential scan is applied during searching, or some indexing technique can be employed. Besides, navigation knowledge is stored at every peer which controls the mechanism of forwarding queries between peers. Peers communicate via messages which are delivered by the underlying computer network. A user can issue a similarity query at an arbitrary peer and the steps depicted in Figure 5.2 are performed to answer the query. First, the peer consults its navigation knowledge to get a list of peers responsible for data partitions that can contain qualifying objects, and forwards the query to them. Since the P2P network can change in time, the navigation can be imprecise, so the query can be forwarded several times

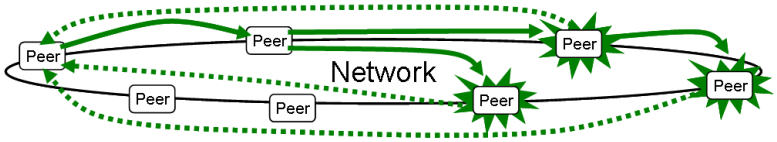

Figure 5.2: A typical query processing in a P2P network.

until it reaches the respective peers (solid arrows). At every peer with a promising partition, a local search procedure is executed that retrieves all objects satisfying the query constraint (peers with star mark). Finally, all the contacted peers return their partial results to the originating peer (dotted arrows), where the final answer is merged and passed back to the user.

So far, we have only considered a single P2P structure that manages data with respect to one modality. Let us now extend this model to embrace multiple modalities $\mathcal{M}_1, \ldots, \mathcal{M}_n$. A complex data object $o \in \mathcal{X}$ is transformed by the projection functions of the respective modalities into a set of descriptors $\{o.f_{\mathcal{M}_1}, \ldots, o.f_{\mathcal{M}_n}\}$. This set together with an identifier of the original object will be denoted as *metaobject*. The similarity of two metaobjects is evaluated by some monotonous aggregated distance function $d^{AG}_{\widehat{\mathcal{M}_1,\ldots,\mathcal{M}_n}}$, e.g. a weighted sum of the partial distances.

To evaluate multi-modal similarity queries efficiently, we build a P2P index for each of the descriptors. So, every single descriptor of a particular metaobject is stored by the respective index along with an identifier of the metaobject. Moreover, a special *zero overlay* is defined where complete metaobjects are stored. The zero overlay allows efficient retrieval of metaobjects using their identifiers as a key – a classical P2P distributed hash table [10] can be used, because we only need the get-by-id operation in this overlay. In principle, these overlays are allowed to share the same infrastructure of physical peers.

Figure 5.3 depicts an image search system with three overlays. The first one is built for image color modality, the second indexes shapes, and the third represents the zero overlay. A metaobject assignment to these overlays is also illustrated in the figure. We can observe that each overlay consists of multiple nodes and their specifics are left up to a particular distributed index structure used in the overlay. These nodes are maintained by physical peers (illustrated by the dotted arrows). Each peer usually manages at least one node from every overlay. Such a mapping is completely transparent for overlay index structures and in general, it is automatically done by the load-balancing mechanism.

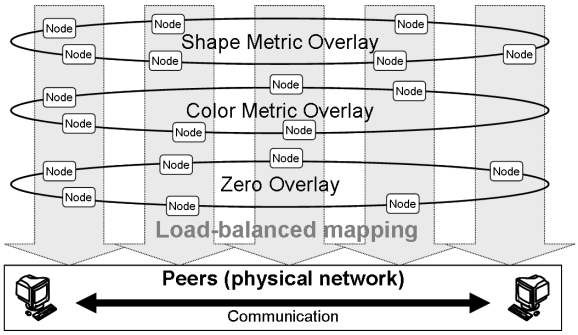

Figure 5.3: Multi-metric overlay setting.

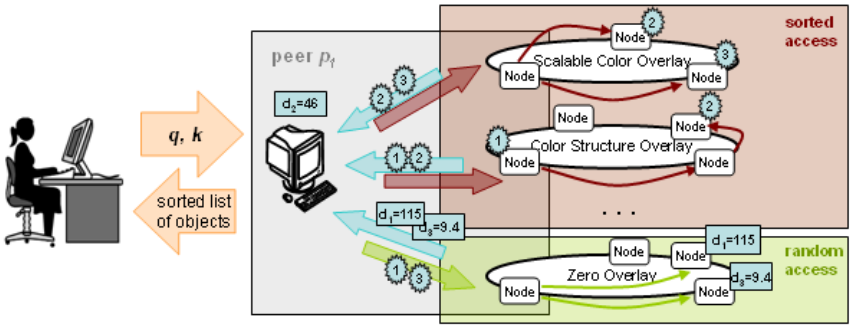

Figure 5.4: Evaluation of a complex query.

Approximate Query Processing

Running a standard TA in a distributed environment would be very expensive, because a single object retrieval is very inefficient. The batch approach is more suitable and it works as follows (see Figure 5.4 for a schematic overview). The issuing peer breaks the query metaobject into its descriptors and executes a nearest neighbor query for every modality in the respective similarity-search overlay. These are evaluated in parallel and a sorted list of the top-most similar objects is returned for each modality. The objects are then used to query the zero overlay to get distances for missing modalities. Next, the user-defined aggregation function is used to compute the objects' overall similarity d_Q. If there are not enough objects with their overall similarity under the actual threshold value τ, the descriptor overlays are requested to provide additional batch of objects until this condition is met.

Unfortunately, a precise evaluation of the Threshold Algorithm can take a lot of time for huge data collections. For example, a 50 nearest neighbor

(50NN) query in a dataset of 1.6 million images takes more than one minute to evaluate even for a batch size of 1,000 objects. Since we know that the interpretation of similarity itself is highly individual and even the optimal search results may not satisfy user needs, it is more reasonable to employ approximate but efficient retrieval. Thus, we alter the stop condition and we end the processing prematurely after ϵ iterations even if the threshold condition is not satisfied. The value ϵ allows us to tune the ratio of the response time and the quality of the result.

Since the parameter ϵ is specific for each query, the system can automatically adjust its value according to user preferences or the actual system load. To help the system tune this parameter more precisely, we can compute actual quality estimations during the iterations of the TA. Since the actual threshold value τ and the maximal aggregated distance in the current result list d^{max} are updated in every iteration of TA, we can see them as functions $\tau(i)$ and $d^{max}(i)$ of the TA iteration i. We can notice two interesting properties of these functions:

1. the precise TA evaluation stops as soon as $\tau(i) \geq d^{max}(i)$ for some i, and

2. $\tau(i)$ only increases while $d^{max}(i)$ only decreases after the result list is filled with k objects.

If we know the final maximal distance $d^{max}(final)$ of the precise result, we can express the quality of the result after i iterations as a ratio $d^{max}(i)/d^{max}(final)$. The best quality is equal to 1 (precise result) while the higher values represent worse quality. However, the final distance is unknown during the evaluation and we can only use the actual threshold value $\tau(i)$ as its lower bound (due to the TA stop condition). The quality is therefore upper-bounded by $d^{max}(i)/\tau(i)$. Using the first ϵ values of $\tau(i)$ and $d^{max}(i)$, we can improve the estimation of the quality by extrapolating the behavior of the $\tau(i)$ and $d^{max}(i)$ functions. Then, their intersection can be computed, and the function value at the intersection is an estimation of the $d^{max}(final)$ and can be used to compute the estimated quality at iteration ϵ.

In the current implementation, we use a linear extrapolation of both these curves. This extrapolation was chosen on the basis of experiments which had shown that for our data collection both curves become nearly linear in later iterations. Another advantage of the linear extrapolation is its low computational cost.

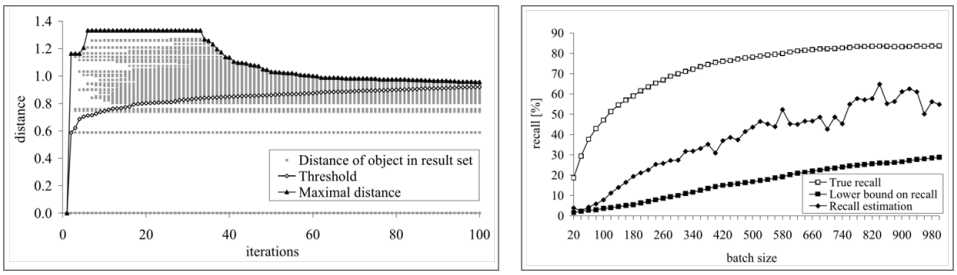

Figure 5.5: Visualization of the standard TA (left), recall of the approximate TA (right).

Experimental Evaluation

For the experiments, we used an instance of MUFIN system loaded with 1.6 million images gathered from the Flickr image gallery [24]. The images were described by the five MPEG-7 descriptors introduced in Section 5.1.3, which were indexed in separate overlays. In order to be able to evaluate results of the distributed Threshold Algorithm, we employed a fixed aggregation function, in particular a weighted sum of the five mono-modal distances as discussed in Section 5.1.3. There were 77 peers in our MUFIN instance, each holding up to five logical nodes of any of the six overlays. These peers were run on a physical infrastructure with 16 CPUs and 64GB RAM in total. For the indexing metric overlays we used the GHT* [20] structure, each node further employed a local M-Tree [47] to actually store the descriptors. For the zero-overlay, we used the Skip-Graphs [10] distributed hash-table.

Standard TA In the first set of experiments, we studied the behavior of the standard TA. In particular, we logged the result sets (i.e. objects in the result set and their distances from the query object) and threshold values computed in individual iterations of TA, and studied their development. The evaluation of a 50NN query is visualized in Figure 5.5 (left) for iterations 1 to 100. Gray squares represent distances of objects in the result set in a particular iteration. The upper dark curve represents the maximal distance $d^{max}(i)$ in iteration i, the lower curve shows the respective threshold value $\tau(i)$. The distance between the curves after each iteration corresponds to the quality of current result set. Our experiments confirmed that the quality increases very quickly in the first few iterations while the rate slows down later.

Distributed TA The second set of experiments evaluated the performance of the distributed TA solution. We executed 50NN queries for 50 randomly chosen query objects, combining all five descriptors using the weighted sum aggregation function. Each query was also evaluated using a sequential scan in order to establish a baseline, i.e. a precise answer to the 50NN query. This was then used for computing recall and distance errors of the approximate search results.

As for retrieval efficiency, the experiments proved that the costs of query evaluation grow linearly with the approximation parameter ϵ (batch size). More specifically, the number of distances evaluations grows sublinearly, but the time saved on distance computations when the batch size increases is spent on maintaining the bigger lists of objects (e.g. sorting) and by the communication between the peers. In terms of effectiveness, we studied the system recall as well as several types of approximation errors. The recall graph is depicted in Figure 5.5 (right) and shows the average recall values as well as the relationships between true result quality, the estimation computed by extrapolation, and the theoretical lower bound on quality provided by the Threshold Algorithm. We can observe that the extrapolation improves the quality estimation quite significantly.

Summary Overall, the experimental evaluation of the approximate TA can be concluded as follows: On average, we can get as good as $80\,\%$ recall in about 4 seconds for 50NN queries, which can be considered acceptable for a standard user. To get a better recall the system would need considerably more time. A more detailed analysis of the experiments can be found in [18].

5.2.2 Content-based Retrieval with Postprocessing

A fundamental requirement that determines the applicability of symmetric fusion techniques is the availability of all modalities to be fused. This is easily satisfied in the previously discussed use case, which exploited different visual descriptors that are always available in an image, but may be problematic for other types of modalities. In particular, let us consider the combination of visual and text features. The text modality is known to provide valuable information about image semantics, but its availability and quality greatly differs for various data sources. In case of applications that need to deal with low text data quality, it may therefore be more advantageous to utilize asymmetric fusion with content-based primary modalities,

which are always available in image collections. The textual information, when present, can be used to refine the basic search results.

Noticeably, asymmetric late fusion solutions that utilize content-based modalities as the primary ones are not as well studied as the inverse case of text-based techniques. Therefore, we focus on exploring the possibilities of content-based searching with text-aware postpocessing of candidate objects in the following sections.

Ranking Phase Fundamentals

As we discussed in Section 4.3.2, we find it convenient to distinguish several phases of the query evaluation process. In this section, we are mainly interested in the result postprocessing part, which we alternatively denote as (re-)ranking. We assume that the set of candidate objects $\mathcal{C}^{BS}$ that is processed in this phase was provided by some content-based search technique employed in the basic search phase.

Let us begin with a formalization of the postprocessing phase, expressing it as a function over the candidate set $\mathcal{C}^{BS}$. A generic function F^{RANK} : $\mathcal{X} \rightarrow \mathbb{N}$ is applied on $\mathcal{C}^{BS}$ to establish a new rank of each object. The actual definition of the ranking function depends on the $context$ in which it is evaluated, which may comprise secondary modalities as well as additional parameters, properties of the candidate set, etc. To improve readability, we relax the strictness of the function definition by including the context parameters in $RANK_{type}$ function as needed. We will discuss the possible context parameters later.

$$F^{RANK}(o, \mathcal{C}^{BS}) = RANK_{type}(o, context) = i,$$

i is the rank of the object $o \in \mathcal{C}^{BS}$ in the given context

The ranking function F^{RANK} needs to satisfy the following *unambiguity condition*:

$$\forall o_1, o_2 \in \mathcal{C}^{BS} : (F^{RANK}(o_1, \mathcal{C}^{BS}) = F^{RANK}(o_2, \mathcal{C}^{BS})) \Rightarrow (o_1 = o_2)$$

Even though users are typically interested in the first k objects with k ranging from 10 to 100, the basic search phase needs to provide significantly more objects to allow the ranking to discover interesting results. The larger the candidate set $\mathcal{C}^{BS}$ is, the higher the chances of finding relevant objects are. On the other hand, if $\mathcal{C}^{BS}$ is too large, the postprocessing step might be too costly. Therefore, the choice of the size $k' = |\mathcal{C}^{BS}|$ needs to balance the following three factors: the costs of the basic search for k' best objects,

the cost of ranking the k' objects, and the probability that there are at least k relevant objects in the initial result of size k'.

In the following, we present several ranking functions that exploit information orthogonal to content-based similarity, which is expected to be used for the basic search. In particular, we endeavour to utilize a textual image annotation when available, and extract semantical information from it. We split the ranking functions into two categories: 1) functions that can automatically rank the initial results, and 2) user-defined ranking where users actively participate in the process of defining the ranking function.

Automatic Ranking

Automatic methods compute the result ranking using only the query context information, i.e. the query object q and eventually some properties of the candidate set $\mathcal{C}^{BS}$. When the candidate set is retrieved by visual similarity, a successful ranking needs to exploit some complementary information available for data objects, e.g. keywords, location, searching object popularity, number of purchases of the object, etc. A more sophisticated ranking can try to identify and exploit some patterns in the properties of objects in the initial result, e.g. the most important keywords, or visual features in case of images. Such approaches are traditionally denoted as pseudo-RF re-ranking. Finally, the ranking phase may also include another type of content-based similarity search. Naturally, several ranking functions can be combined to provide the final order of objects.

In the following, we focus on text-based automatic ranking in collections with annotations of various quality, which is common in many web applications such as photo galleries.

Keyword ranking Inversely to the search model applied by common web search engines that combine text-based retrieval and visual ranking, we propose to rank the content-based search result with respect to keywords of the query image. The keywords need to be available for the image, which is typically satisfied when the query object q is taken from $\mathcal{X}$ (e.g. in a collection browsing scenario). The similarity between two sets of keywords is measured by the Jaccard coefficient (see [165] for a formal definition of the Jaccard similarity).

$$RANK_{queryObjectKeywords}(o, \mathcal{C}^{BS}, q) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | (d_{Jaccard}(q.words, y.words) < d_{Jaccard}(q.words, o.words))\}$$

This ranking method is intended for queries with rich and reliable annotations. In order to broaden the ranking range, we apply stemming and use the WordNet lexical database [69] to retrieve additional keywords from semantic relationships, as suggested in study [102]. We also remove all words that are not nouns, verbs or adjectives.

Word cloud ranking For queries with none or sparse and erroneous text metadata, the keyword ranking is not applicable. In this case, we propose to exploit the keywords of all objects in $\mathcal{C}^{BS}$. The keywords are first cleaned and expanded by WordNet as described above. Then we compute the frequencies of the keywords from all objects in $\mathcal{C}^{BS}$. We denote the resulting set of keywords with their respective frequencies as *word cloud*. Finally, the ranking employs the n most frequent words from the cloud (denoted as $wordCloud(\mathcal{C}^{BS}).top(n)$) as the query object words in the text-similarity evaluation. Noticeably, q does not play any role in this ranking, which exploits the pseudo-RF approach.

$$RANK_{wordCloud}(o, \mathcal{C}^{BS}, n) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | (d_{Jaccard}(wordCloud(\mathcal{C}^{BS}).top(n), y.words)$$
$$< d_{Jaccard}(wordCloud(\mathcal{C}^{BS}).top(n), o.words))\}$$

Combined visual and text ranking In the previous methods, we have only used the textual (keyword) information for the ranking, ignoring the initial ranking of the visual (content-based) search. As we discussed in Section 4.4.7, this follows the typical pattern of asymmetric fusion solutions, where the primary modality is used only in the basic search phase and the re-ranking is performed by the secondary modality. However, the initial result is retrieved using the kNN operation which provides the ranking of its own, and it may also be useful to factor this into the final ranking. Therefore, we enrich the $RANK_{queryObjectKeywords}$ method by summing the text-induced distance with the distance of the respective object from the visual space, defined by some primary modality $\mathcal{M}_{visual} = (d_{vis}, p_{vis})$. Since the Jaccard measure gives values between zero and one, we need to normalize d_{vis} so that both of the two summed distances influence the ranking equally. Thus, we multiply d_{vis} by a normalization factor f.

$$RANK_{queryObjKwAndVisual}(o, \mathcal{C}^{BS}, q, f) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | (d_{Jaccard}(q.words, y.words) + f \cdot d_{vis}(p_{vis}(q), p_{vis}(y))$$
$$< d_{Jaccard}(q.words, o.words) + f \cdot d_{vis}(p_{vis}(q), p_{vis}(o)))\}$$

Adjusting the factor f can also be used to strengthen or diminish the impact of the visual descriptors on the ranking. Moreover, the $RANK_{wordCloud}$ can be modified in a similar fashion resulting in the $RANK_{wordCloudAndVisual}$ function that combines the results of the word cloud ranking with the visual distances.

Adaptive keyword/cloud ranking For datasets with highly variable quality of text metadata, it can be beneficial to choose the ranking method adaptively. Therefore, we propose the following heuristic that combines the previous raking methods. Given the query object's keywords and the word cloud obtained from $\mathcal{C}^{BS}$, we prepare the set of adaptive keywords A as follows. First, all the cleaned keywords of the query object are inserted. If there are less than c of these, the most frequent cloud words are added. However, the cloud words must exhibit some minimal frequency t to be considered relevant. Note that the WordNet cleaning and enrichment as defined above is used. The final ranking is computed as a combination of the text ranking defined by the described keyword set and the initial visual ranking.

$$A := q.keywords \cup \mathcal{C}^{BS}.wordCloud.top(c - |q.keywords|, t)$$

$$RANK_{adaptive}(o, \mathcal{C}^{BS}, q, c, t, f) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | (d_{Jaccard}(A, y.words) + f.d_{vis}(p_{vis}(q), p_{vis}(y)) <$$
$$d_{Jaccard}(A, o.words) + f.d_{vis}(p_{vis}(q), p_{vis}(o)))\}$$

User-Defined Ranking

As we already know, the understanding of similarity is subjective and varies in different conditions. Therefore, it is not always possible to obtain the optimal result automatically and a user cooperation is required. In this case, the system can display the results of the initial search and require additional user input for the ranking phase. A new query object, a measure of the relevance of the initial result, or a specification of relevant values for associated object metadata are a few examples of possible user input for the ranking phase.

While the user-defined ranking functions can be very powerful, they need attention, knowledge, and time from the user. Therefore, these are only intended as advanced options for more experienced users. In the following subsections, we define two ranking functions for advanced searching in image data with text annotations.

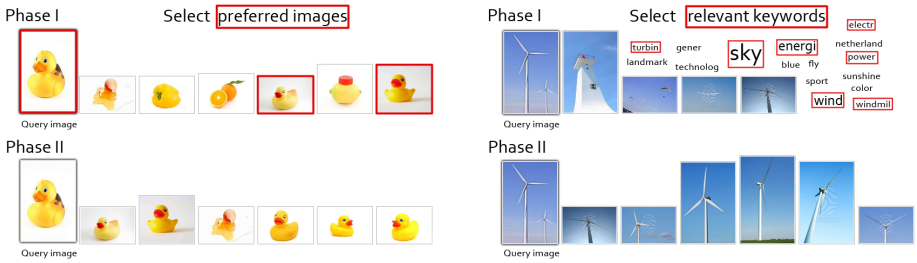

Figure 5.6: User-defined ranking: relevance-feedback ranking (left), user-defined text ranking (right).

Relevance feedback ranking In some search systems, users can provide a feedback on the relevance of results and ask for a refined result. To provide this, the system uses the relevance information to modify the query object or the similarity measure (see [168] for more details). This may be repeated in several iterations which finally produce a better result but may take a considerable amount of time, as a new query needs to be evaluated in each iteration. Therefore, we propose to implement the relevance feedback as the ranking function. As depicted in Figure 5.6, users can choose relevant objects from the initial result. The ranking function then evaluates the final rank of object $o \in \mathcal{C}^{BS}$ as a function of the content-based similarity d_{vis} between o and each of the objects marked as relevant.

$$RANK_{RF}(o, \mathcal{C}^{BS}, d^{AGG}, [q_1, \ldots, q_n]) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | d^{AGG}(d_{vis}(p_{vis}(q_1), p_{vis}(y)), \ldots, d_{vis}(p_{vis}(q_n), p_{vis}(y)))$$
$$< d^{AGG}(d_{vis}(p_{vis}(q_1), p_{vis}(o)), \ldots, d_{vis}(p_{vis}(q_n), p_{vis}(o)))\}$$

In general, no limitations are imposed on the aggregation function d^{AGG}. The most natural choices are however monotonic functions such as SUM, MIN or MAX.

User-defined keyword ranking Keywords may provide a strong ranking tool but automatic approaches may not always guess the optimal set of words. Therefore, another user-defined method allows users to select the relevant keywords of their own choice.

$$RANK_{userKeywords}(o, \mathcal{C}^{BS}, userWords) = i \in \mathbb{N}, i = |Y|,$$
$$Y = \{y \in \mathcal{C}^{BS} | (d_{Jaccard}(userWords, y.words)$$
$$< d_{Jaccard}(userWords, o.words))\}$$

One way of using this type of ranking is to let the users type any keywords they consider relevant. However, there is a high possibility that their choice will not match the keywords used in the images' metadata. Therefore, we allow the users to choose from the list of keywords contained in the initial result.

Experimental Evaluation

To evaluate the quality of all the ranking functions we proposed, we organized several user-satisfaction surveys. Using a simple web interface, the participants were shown the initial and the ranked result sets and had to mark the relevant objects. In the two cases of user-defined ranking, the participants were first asked to choose the relevant objects/words from the initial result and then they evaluated the new ranking. About 40 users of different age, sex and computer skills participated in the experiments.

For the experiments, we used two different datasets. Dataset 1, which comes from a commercial microstock site, contains 8.3 million high-quality images with rich and systematic annotations (about 25 keywords on average). Dataset 2 contains 100 million images from the Flickr web site [24] and exhibits worse quality of images and sparse and erroneous keywords. In each set of experiments, we used 50 randomly chosen query objects. For an easy visualization of several result sets on a screen, we only used a result set with 10 objects. In the initial nearest neighbor search we always retrieved 200 objects, which were conveyed to the ranking function. The relevance of result is measured as a user-perceived precision, i.e. the ratio of the number r of objects marked as relevant to the number t of all displayed objects from the result.

Apart from evaluating the performance of individual postprocessing methods, we also used the experiments to study the usefulness of the ranking in principle. For each result, we asked users whether they want to try re-ranking. About 50 % of results over Dataset 2 (the worse one) and 72 % of results over Dataset 1 were considered worth trying; the rest of the result sets was either perceived as already very good (17 % for Dataset 1) or too bad (33 % for Dataset 2). In case of Dataset 2, we remark that the low quality of results was caused by the low quality of some of the randomly picked query images rather than bad performance of the basic search.

Relevance-feedback ranking First, we experimented with user-defined ranking methods in order to get some insight into user preferences. The $RANK_{RF}$ method was evaluated over both datasets, using SUM as the dis-

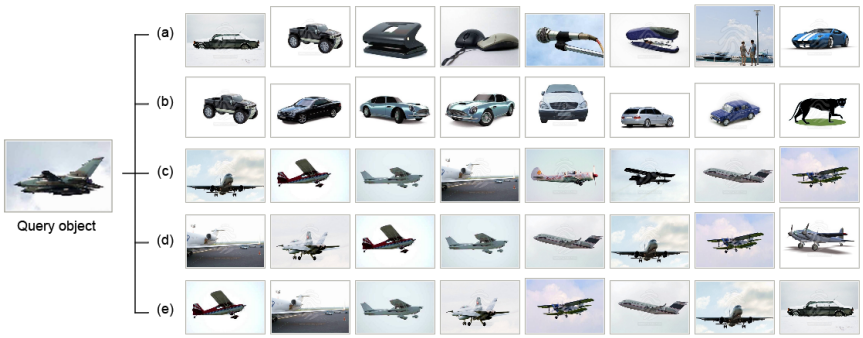

Figure 5.7: Automatic ranking: (a) simple visual-based search, (b) rank by word cloud – size 5, (c) rank by word cloud – size 10, (d) rank by query object keywords, (e) rank by query object keywords and visual distance.

tance aggregation function. For Dataset 1, we also processed exact *multi-object queries* that retrieve the k objects from $\mathcal{X}$ that are most similar to a set of query objects. Obviously, the relevance-feedback ranking provides an approximate result for a multi-object query.

The experiments revealed that for both datasets, the precision of results was increased by approximately $20\,\%$ using the $RANK_{RF}$ ranking. The precision of results obtained by the multi-object query was only slightly better that the precision of re-ranked results. This confirms our assumption that there are enough good objects in the candidate set and re-ranking approaches can be highly effective.

Text ranking The ranking based on users' choice of keywords was evaluated only for Dataset 1. Participants of the experiment were shown the initial result and a set of keywords, which comprised all keywords of the query object combined with the 50 most frequent keywords from the word cloud. Different font sizes were used to emphasize the most frequent keywords, as depicted in Figure 5.6 (right). Users were asked to choose any number of relevant keywords and evaluate the re-ranked result.

The results showed that the keyword-based ranking increases user satisfaction by 15 %. On average, users selected 3-4 words per search. The collected data also indicate that the more keywords were issued, the higher the satisfaction with the result was. About 90 % of all keywords selected by users belonged to the query object keywords. This confirms our assumption that text metadata in Dataset 1 are of a high quality.

	Dataset 1 result precision	Dataset 2 result precision
simple content-based search	36.2 %	23.5 %
$RANK_{wordCloud}(o, \mathcal{C}^{BS}, q, 5)$	33.2 %	25.4 %
$RANK_{wordCloudAndVisual}(o, \mathcal{C}^{BS}, q, 5, f)$	41.3 %	32.5 %
$RANK_{wordCloud}(o, \mathcal{C}^{BS}, q, 10)$	35.1 %	24.9 %
$RANK_{wordCloudAndVisual}(o, \mathcal{C}^{BS}, q, 10, f)$	42.0 %	33.7 %
$RANK_{queryObjectKeywords}(o, \mathcal{C}^{BS}, q)$	55.4 %	41.1 %
$RANK_{queryObjKwAndVisual}(o, \mathcal{C}^{BS}, q, f)$	56.8 %	43.0 %
$RANK_{adaptive}(o, \mathcal{C}^{BS}, q, 10, 10, f)$	56.8 %	45.4 %

Table 5.2: Performance of automatic ranking methods.

Automatic ranking Another set of experiments was designed to test the performance of the proposed automatic ranking methods over both datasets. Participants of the experiments were shown several sets of results on one page and asked to mark the relevant ones. Figure 5.7 shows a part of one such screen.

Some of the automatic methods are further specified by parameters. In particular, the $RANK_{wordCloud}$ and $RANK_{wordCloudAndVisual}$ functions may work with a variable number of most frequent words. In the experiments, we tested two values of this parameter to understand its influence on the quality of results. The values 5 and 10 were chosen using our experience from the user-defined ranking.

Table 5.2 comprises the obtained statistics. Clearly, the best results for Dataset 1 are achieved when the keywords of the query object are taken into consideration. This observation conforms to the conclusion we derived from the user-defined ranking experiments. The adaptive ranking technique used the same keywords as the $RANK_{queryObjKwAndVisual}$ most of the time. As for Dataset 2, the query object keywords cleaned and enriched by the WordNet achieve 10 % improvement of result quality. However, the best results were obtained by the adaptive ranking which capitalized on the cloud information combined with query object keywords.

Processing time As one of our objectives is providing effective and efficient processing of large datasets, we also studied the relationships between the obtained quality and the computation costs. The basic search phase ex-

ploited a scalable and efficient MUFIN infrastructure, the average response time of the initial search being 500 ms. The ranking phase costs naturally depend on the number of processed objects. For a candidate set of 200 objects, the average time needed for post-processing was about 30 ms. Let us recall that the postprocessing provides results of a quality comparable to the results of the multi-object query, which guarantees precise results. However, the costs of a precise evaluation of the multi-object query is much higher, ranging from seconds to tens of seconds.

Summary Our experiments show that the combination of the content-based retrieval with postprocessing methods can improve the satisfaction of users significantly. The most successful ranking methods nearly doubled the user-perceived quality of the results. An additional analysis of experimental results has also discovered that even though the query result set still contains some irrelevant objects, the most relevant ones were pushed to the top. The performance of the ranking methods depends heavily on the relevance of data objects in the initial result set. In the experiments, we verified our assumption that there is a significant amount of relevant objects in the enlarged result of a general content-based search that are scattered among other objects and thus do not appear on the first result page. When several hundred top-ranking objects are submitted to the ranking method, the final result is comparable to the result of a much more expensive query processing over the whole dataset, which we could observe for the relevance feedback ranking.

5.2.3 Inherent Fusion

The key factor of the asymmetric postprocessing fusion is the fact that the candidate set $\mathcal{C}^{BS}$ that is passed to the ranking phase has a preset size. If this size is too large, the search performed on the primary modality can be very costly. Moreover, the candidate set has to be fully enumerated, which may require additional memory and communication costs. On the other hand, if $\mathcal{C}^{BS}$ is too small, the result will be strongly affected by the primary modality, since the objects that would be considered highly relevant by the secondary modality are unlikely to appear within the candidates.

We believe that it is possible to significantly improve the performance of asymmetric solutions, if we more thoroughly exploit all information available during the query evaluation. In the following, we introduce the *inherent fusion* technique, which allows to implement the asymmetric fusion in a more efficient and scalable way.

Technique Introduction

Let us first have a closer look at the processing in the basic search phase. In general, indexing techniques typically partition the dataset into a number of, not necessarily disjoint, data chunks (intervals, areas, clusters, posting lists, etc.). Let us denote these partitions $P_1, \ldots, P_n$, where $P_i \subseteq \mathcal{X}$. During the evaluation of a query, the index typically prunes some of these partitions and accesses the potentially relevant data in the rest of them to select the answer set. The number of objects accessed in this way is significantly (orders of magnitude) larger than the typical number of candidates in $\mathcal{C}^{BS}$.

Following this observation, we designed the inherent fusion technique so that it utilizes all objects visited by the index also for ranking by secondary modalities. Similarly as in the asymmetric fusion techniques, we propose to index the data using selected primary modalities $\mathcal{M}^P$ but store the full data objects, so that all primary and secondary modalities ($\mathcal{M}^P$ and $\mathcal{M}^S$) are held by the index. At query time, the index processes all objects from all non-pruned partitions; let us denote these objects as *super-candidate set* defined as $\mathcal{C}^S = \bigcup_{i=1}^{m} P_i$, where P_i is a data partition that cannot be pruned for the query object q. Instead of a standard accumulation of the best-seen objects with respect to the primary modality $\mathcal{M}^P$, each object from $\mathcal{C}^S$ is evaluated by the multi-modal similarity function d_Q which takes into account both the primary and secondary modalities.

The difference between the standard asymmetric postprocessing fusion and inherent fusion is schematically shown by Figure 5.8. In case of the postprocessing (left schema), the index on $\mathcal{M}^P$ identifies relevant partitions P_i and ranks the data from these partitions by $d_{\mathcal{M}^P}$ to create the candidate set for further processing by the query distance d_Q. As described above, the inherent fusion (Figure 5.8, right) ranks the objects directly by d_Q as they are accessed in the partition P_i. In comparison with the postprocessing fusion, the volume of data searched with all modalities ($|\mathcal{C}^S|$) is considerably larger, increasing the probability of discovering more relevant objects. Importantly, this processing is far less costly than the asymmetric postprocessing fusion on a candidate set of the same volume because that would typically require processing of even larger $\mathcal{C}^S$ within the index on $\mathcal{M}^P$. Obviously, it is not guaranteed that $\mathcal{C}^S$ contains the $|\mathcal{C}^S|$ best objects with respect to $\mathcal{M}^P$; on the other hand, this approximation allows us to keep the processing costs low.

The inherent fusion can be implemented relatively easily within most of the standard indexing techniques. First, the index needs to store the data for the secondary modalities $\mathcal{M}^S$ so that the ranking can be applied, but

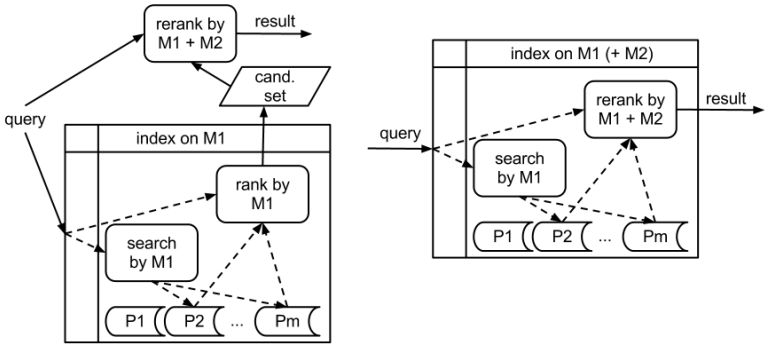

Figure 5.8: Schema of asymmetric postprocessing fusion (left) and inherent fusion (right).

storing additional (raw) data is usually supported. Second, the query evaluation procedure needs to be modified, so that additional computation can be added to the part where the resulting set is accumulated during the processing. This might be possible to register via hooks (callback methods), if the implementation allows it, or the code must be modified accordingly. Finally, the system must be modified so that the query primary and secondary modalities are split and passed to the original index partition traversal or the modified result set accumulator respectively. In our implementations, we utilize the MESSIF library [21], which contains all the necessary support, so any index structure implemented on top of MESSIF can transparently take advantage of the inherent fusion.

The inherent fusion is a straightforward extension of the re-ranking paradigm, however the advantages obtained by this solution are considerable. We review them with respect to the standard quality measures of retrieval methods:

- flexibility: similarly to the standard re-ranking, there are no requirements on the way in which modalities $\mathcal{M}^P$ and $\mathcal{M}^S$ are combined at query time (for instance, arbitrary weighting) and there are no limitations on the indexability of the additional modalities ($\mathcal{M}^S$);
- effectiveness: relatively large set of objects can be probed with all modalities, which is likely to improve the quality of the results;
- efficiency: the whole evaluation is done within the index without explicit enumeration of the whole candidate set $\mathcal{C}^S$ and without any data replication, which allows us to keep the processing costs low;

- scalability: this approach allows efficient exploitation of distributed indexing techniques; scalability of the index structure is thus straightforwardly exploited to guarantee also the scalability of the inherent fusion.

The only disadvantage of the inherent fusion solution that we are aware of is the fact that the data stored in the index must contain also the secondary modalities, so the index requires more storage space. On the other hand, the secondary-modality data needs to be stored in any case for the postprocessing phase, as an online extraction of the respective descriptors would be too costly. The difference is thus only in the implementation of the storage facilities.

Experimental Evaluation

The performance of the inherent fusion technique was evaluated in a set of experiments over a real-world dataset. In particular, we employed the Profiset collection, its test objects and user-evaluated ground truth, all of which will be discussed in more detail in Chapter 6. The Profiset collection contains 20 million high-quality images with rich keyword descriptions, each of the 100 test queries was defined by an example image and one or several keywords.

In the experiments, we compared different ways of asymmetric fusion of a primary visual and secondary textual modality. The visual similarity of images was evaluated by the fixed combination of five MPEG-7 descriptors introduced in 5.1.3, the textual similarity was computed by the cosine distance with *tf-idf* weighting. The overall distance measure d_Q was computed as a sum of the normalized visual- and text-based distances. Dataset objects were indexed by the visual similarity, employing a centralized M-index structure [124] which was adapted to accommodate the inherent fusion.

To assess the performance of the inherent fusion, we employed the following three quality measures: the objective result quality, as measured by the distance function; the subjective result quality as perceived by users; and the query processing costs measured by wall-clock time. For comparison, we also measured the performance of a standard re-ranking solution and a precise Threshold Algorithm, which represent the theoretical lower and upper bound, respectively, on results quality as well as processing costs.

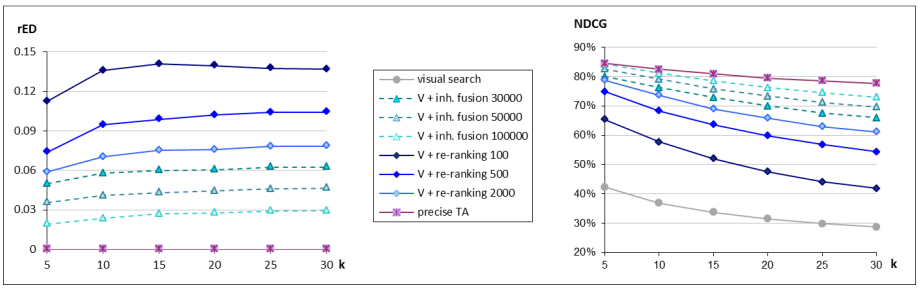

Figure 5.9: Inherent fusion: Average distances at a given rank (left), average NDCG at a given rank (right)

Distance-based result quality A distance-based evaluation of result quality is an objective method of effectiveness assessment that compares the result of a given approximate search technique R^A to a precise retrieval result R^P. From several commonly used distance-based quality measures [165], we chose the *relative error on distance at k*, which compares the distances of objects at k-th position (d^k) in approximate and precise results: $rED(k)_{R^A} = d^k_{R^A}/d^k_{R^P} - 1$.

Figure 5.9 (left) plots the rED curves of different re-ranking and inherent fusion methods as well as the zero error line for the Threshold Algorithm. For the approximate fusion methods, the number in the method label represents the size of candidate set which enters the fusion phase. We can observe that for all asymmetric processing methods, the error continues to decrease with the growing size of $\mathcal{C}^{BS}$ and $\mathcal{C}^{S}$. However, it is important to notice that the dependence between the objective quality and the candidate set size is approximately logarithmic. Further improvements of result quality would thus require processing significantly more objects with respect to d_Q, which confirms the need for efficient fusion techniques.

User-perceived quality The second quality evaluation takes into account the user-decided relevance of results. In an ideal case of a perfect distance function that precisely captures user's information need, the user-perceived quality would copy the distance-based evaluation. In reality, however, these two perspectives may significantly differ. To be able to study the users' opinion on result quality, we organized an evaluation campaign described in Chapter 6 and collected relevance assessments for individual queries and result objects. We then measured the user-perceived quality by the *normalized discounted cumulative gain at k* (NDCG(k)) met-

	1 CPU	8 CPU
visual search	587 ms	415 ms
V + re-ranking 2000	789 ms	616 ms
V + inherent fusion 30000	590 ms	430 ms
V + inherent fusion 50000	852 ms	603 ms
V + inherent fusion 100000	1474 ms	999 ms

Table 5.3: Inherent fusion: Average processing time.

ric [90], which considers the relevance scores of result objects on positions 1 to k, giving more weight to higher ranking results.

The NDCG quality measure confirms that the inherent fusion technique improves the result quality, as can be seen in Figure 5.9 (right). Again, the dependence between the candidate set size enlargement and the quality improvement is logarithmic.

Efficiency To assess the efficiency of the inherent fusion, we also measured the wall-clock time needed for evaluation of a single query. In order to obtain the baseline costs of each method, we first run our experiments using only one CPU. Then, we performed another set of experiments that utilized 8 CPUs and thus allowed the M-index structure to utilize its internal parallelization, which is also exploited by the inherent fusion technique. The average costs of inherent fusion methods are summarized in Table 5.3, which also provides a comparison with the costs of a simple visual-based search and a standard re-ranking fusion solution.

Summary The presented results show that the inherent fusion technique achieves a very good performance-costs trade-off over real-world data. Using this technique, we are also able to provide a scalable multi-modal solution, since the scalability of underlying index structure is straightforwardly exploited.

5.3 Large-Scale Evaluation of Multi-Modal Retrieval

As we have debated in Section 4.3.5, it is nearly impossible to assess the performance of a given retrieval solution theoretically. We can separately analyze the properties of different modalities, extraction algorithms, indexing techniques, or fusion models, but it is very difficult to model the whole

retrieval process and the ways in which one technique influences the performance and effectiveness of another. At the same time, however, it is very important to know which solution is suitable for which task and what conditions influence the suitability. When a theoretical analysis is not sufficient, a complementary view on the problem can be provided by an experimental evaluation of the precision-costs tradeoff in various settings. Of course, the experiments need to be carefully designed so that the results for individual methods are comparable and the comparison brings relevant and reliable information.

Although experimental evaluations are contained in most research works that introduce novel retrieval techniques, the available comparisons are only partial. For instance, a re-ranking method that utilizes a text-based basic search is typically compared to a simple text search and eventually to other re-ranking techniques, but there is no comparison of behavior of the re-ranking approach and the Threshold Algorithm. We could observe this phenomenon also in the evaluations reported in the previous section. Moreover, each evaluation is likely to utilize a different dataset, queries, and performance measures. While natural, these facts prevent a more thorough understanding of suitability of individual techniques for different tasks.

In this section, we try to address this problem by providing a comprehensive experimental evaluation of methods for large-scale searching in images with a late modality fusion. Clearly, it is not possible to implement and compare all solutions that have ever been proposed. To keep the task feasible, we only consider basic modalities and the fundamental search strategies. We believe that such evaluation is much needed and will lay foundations for future more advanced analyses. In the following, we formulate several specific questions we would like to answer in our evaluation, discuss the selection of methods to be compared, describe the evaluation process, and present some of the most interesting results.

5.3.1 Objectives

Our comparative analysis addresses various aspects of late fusion methods that combine the two most popular modalities of image retrieval, i.e. text and visual features. Recent research activities as well as commercial applications have shown that this is a very promising way to achieving flexible and efficient searching in web-scale image retrieval. Obviously, there are many interesting questions related to visual-and-text image retrieval that would be worth attention and could be studied in real-world usage simulations. In our research, we decided to focus on discovering the fundamental

characteristics of various approaches to late modality fusion, and their behavior on different data collections. The following sections define the particular objectives of our comparative analysis and outline the methodology of the experimental evaluation. All following problems assume a web-like search scenario where a multi-modal query has been provided by the user.

Problem 1: *Efficiency and effectiveness of late-fusion paradigms in large-scale*

In the survey of multi-modal search techniques presented in Section 4.4, we have categorized the efficiency of different approaches into several classes, taking into account the amount of data that needs to be processed. However, this theoretic classification gives us little information about the real costs of a given solution, which often depend also on the distribution of the data in the searched space. It is thus possible that the actual efficiency of the Threshold Algorithm is good, even though its theoretical maximum costs are very high.

To improve the efficiency and scalability of searching, many late fusion techniques also utilize some level of approximation during query evaluation. In the proposal of most such techniques, it is demonstrated that the quality loss of the respective solutions is acceptable. However, it is hardly possible to compare the retrieval precision of several approximate methods since different query scenarios and data collections were used in the reported experiments.

Approach We implement different late fusion methods in a uniform environment, process a batch of selected queries over a suitable large-scale dataset, and measure the query evaluation costs as well as the relevance of results. Moreover, we study the influence of approximation parameters (in particular the size of $\mathcal{C}^{BS}$) on these qualities.

Problem 2: *Influence of dataset quality on late fusion techniques performance*

When we perform the evaluation outlined in approach to Problem 1, we obtain some assessment of performance of given search methods. These results should be relevant for datasets that have similar properties as the one that was used in the evaluation. However, the real-world databases tend to differ significantly in the quality of data they provide for individual modalities. Clearly, it would be helpful to know which trends in retrieval performance are dependent on the characteristics of the particular dataset and which are not.

Approach We select a second large-scale dataset with characteristics significantly different from the first one, evaluate the selected retrieval methods over both of these collections, and analyze the differences between the relevance data we obtain from the experiments.

Problem 3: *Query-level analysis of retrieval performance*

Apart from measuring the average performance of selected retrieval methods, we would also like to examine the retrieval behavior of individual queries. For further development of search techniques, it would be useful to know which approaches are suitable for which types of queries and how the individual query types can be recognized. Obviously, this is a very difficult task and we do not expect to find the final answer. On the other hand, the large amount of experimental results allows us to search for some trends in retrieval behaviour as well as potential relationships between different characteristics of queries, answer sets, and the suitability of individual retrieval techniques.

Approach In our experiments, we log the distances and ranks of individual objects in answer sets. Then, we search for dependencies between these characteristics and the suitability of selected search methods for a given query. We also confront the two sets of results obtained from different data collections to discover the influence of the dataset characteristics on the retrieval behavior.

Problem 4: *Usefulness of query expansion for large-scale image search*

One of the frequently discussed possibilities of improving the retrieval quality is automatic query expansion, i.e. an automatic refinement of the query $Q = (q, d_Q)$ performed prior to the actual query evaluation. In most solutions when this strategy is exploited, the text modality is subject to some expansion procedure that utilizes different vocabularies and ontologies to identify additional keywords. However, the reported results are uncertain: in some cases, result quality improvement is claimed, but in other reports, the query expansion introduces noise. Evidently, the noise can be easily brought into retrieval e.g. when the expansion does not correctly determine the meaning of some ambiguous term. Unfortunately, finding the correct meaning of a given query keyword is often a difficult task that requires costly processing of different information sources. A question arises whether it is worth investing efforts into the expansion, i.e. whether the

possible retrieval precision improvement is worth the increased costs of the query processing.

Approach We evaluate selected methods with expanded query keywords and compare the relevance of results obtained for original and expanded queries. To eliminate some of the noise, we do not employ automatic query expansion but manually link the query keywords to relevant objects (synsets) in the WordNet lexical database [69]. The synonyms in this synset and related synsets determined by WordNet relationships are used to expand the query. This way, we obtain an estimate of relevance improvement that can be gained by WordNet-based expansion, which is the most frequently used approach in the query expansion field.

5.3.2 Selected Retrieval Techniques

As anticipated, we limit our study to the two modalities most frequently used in image retrieval. In particular, we employ a text similarity of keyword image descriptions and a global visual similarity of image content. The text similarity is expressed by the cosine distance and standard *tf-idf* weighting schema [12], whereas the visual similarity is evaluated by a static combination of selected MPEG-7 descriptors [67]. As for the selected search methods, we are particularly interested in the comparison of precise and approximate solutions with different approaches to the integration of modalities, and the differences between text-based and visual-based solutions in case of asymmetric fusion scenarios. With respect to these objectives, we selected the following methods for the experimental comparison:

- baseline solutions: text-based retrieval, content-based retrieval;
- symmetric basic-search fusion: precise Threshold Algorithm (described in Section 4.4.5);
- symmetric postprocessing fusion: approximate Threshold Algorithm with a fixed size of $\mathcal{C}^{BS}$ (described in Section 5.2.1);
- asymmetric basic-search fusion: text-based retrieval with inherent fusion, content-based retrieval with inherent fusion (described in Section 5.2.3);
- asymmetric postprocessing fusion: text retrieval with visual-only or multi-modal re-ranking, content-based retrieval with text-only or multi-modal re-ranking.

To guarantee a fair comparison of the selected techniques, all of them were implemented in a uniform environment of the MESSIF framework for similarity searching [21]. In particular, the M-index structure [124] was employed to support the content-based retrieval, and the Lucene engine [111] was utilized for the text-based searching. For approximate solutions, several settings of the sizes of $\mathcal{C}^{BS}$ and $\mathcal{C}^{S}$ were tested to discover the dependence of result characteristics on these parameters. Furthermore, we also compared the two possible approaches to asymmetric postprocessing fusion discussed in Section 4.4.7 – the mono-modal re-ranking, in which only the secondary modality determines the ranking of candidate objects and subsequent selection of the final answer set, and the multi-modal re-ranking, where both the primary modality and the secondary modality influence the final ranking of candidate objects.

5.3.3 Evaluation Methodology

To test the performance of methods intended for large-scale retrieval, it is necessary to perform the evaluations over large datasets with real-world data. In this section, we introduce the data collections we selected and justify our choice of performance measures.

Datasets, Queries and Ground Truth

The performance of a search method is significantly influenced by the input data – a text-based retrieval is likely to perform well on a collection with good annotations, but poorly otherwise. Therefore, we decided to include two different data collections in the evaluation. The first of them is the Profiset collection, which was specially created for our evaluations as described in Chapter 6. The Profiset collection contains 20 million stock photos with rich and precise keyword annotations. For the second evaluation we employ a 20 million subset of the CoPhIR collection [24] consisting of Flickr images and tags, which are of a lower quality.

To evaluate the retrieval quality, we defined a set of 100 queries, each of which is composed of an example image and a short description. The topics comprise a selection of the most popular queries from search logs provided by a commercial partner, and several queries that are known to be either easy or difficult to process in content-based searching. More information about the queries can be found in Section 6.3.2. A 30NN query was evaluated for each of the studied methods and each query object.

One of the result relevance metrics we want to study is the user satisfaction with results. To be able to measure this, we need a ground truth for the given queries and collections, i.e. a set of user-provided relevance assessments for each result. In Section 6.3.3 we describe how we collected the ground truth data for our queries. Each of the results was evaluated by at least two human judges, who marked it as *highly relevant*, *partially relevant*, or *irrelevant*. These categories were then transformed into relevance percentage and averaged. Noticeably, our ground truth is only *partial* – we have relevance assessments for all results returned by any of the methods under comparison, but we do not know the complete set of relevant objects for each query.

Performance Measures

To evaluate the overall performance of individual retrieval methods, we need to compare both their costs and the quality of results. Concerning the search efficiency, it is most natural to measure the wall-clock time of query evaluation, since all the methods utilize the same implementation framework and hardware. The experiments were run on a single machine with 8 CPU cores and 32GB RAM. In order to obtain the baseline costs of each method, we have first run the experiment using only one CPU. In the other set of experiments we have used all 8 CPUs and thus allowed the indices to utilize their internal parallelization.

As for effectiveness, we apply two different views. The first one is used for approximate techniques and compares their results to the precise answer, provided by the Threshold Algorithm. Let R^A be an approximate result and R^P the precise answer. The *relative error on distance at k* takes into account the distances of objects k-th position (d^k) in the respective results: $rED(k)_{R^A} = d^k_{R^A}/d^k_{R^P} - 1$.

The other result quality measure takes into account the user-perceived relevance. This evaluation metric needed to be chosen carefully as our ground truth data is not of the typical sort assumed in information retrieval – it is incomplete and with non-binary evaluations of relevance. Results quality measures used in information retrieval are nicely analyzed in [109]. For kNN queries with ranked results, the authors discuss the following measures:

- *Recall* and *precision* are two basic relevance measures for unranked sets. In case of ranked results, we can consider the *precision-result curve* which plots the dependence between values of these two metrics. This curve can be also transformed into a single number – the

Mean Average Precision (MAP), which is computed as the average precision of different recall levels. Noticeably, the complete ground truth needs to be known in order to evaluate recall.

- *Precision at* k is a measure suited to large-scale retrieval applications, where the recall is not as relevant to users as the precision of results that appear on the first k positions. This measure does not require any estimate of the number of relevant documents. However, it is the least stable of the commonly used evaluation measures and it does not average well, since the total number of relevant documents for a query has a strong influence on precision at k.

- *R-precision* measure requires having a set of known relevant documents Rel, from which we calculate the precision of the top Rel documents returned. Noticeably, the set Rel may be incomplete. R-precision adjusts for the size of the set of relevant documents, averaging this measure across queries thus makes more sense. Interestingly, the R-precision is equal to both precision and recall of Rel.

- *Cumulative gain*, and in particular *normalized discounted cumulative gain* (NDCG) [90] is another approach that has seen increasing adoption in ranked results evaluation. NDCG is designed for situations of non-binary notions of relevance. Like precision at k, it is evaluated over some number k of top search results. For a set of queries Q, let $R(j, d)$ be the relevance score assessors gave to document d for query j. Then,

$$\mathrm{NDCG}(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)} - 1}{\log_2(1+m)},$$

  where Z_{kj} is a normalization factor calculated so that a perfect ranking's NDCG at k for query j is 1.

In our evaluation, we do not have a full ground truth and the partial ground truth is collected after the experiments are evaluated. Therefore, recall, MAP and R-precision measures cannot be used. The NDCG measure, on the other hand, is very well suited to our data. We can also adjust the Precision at k measure to consider the non-binary relevance. Both these measures allow us to evaluate the precision of the top k results retrieved by a particular method relatively to the best known results for the given query, with the former also taking the ranking of result objects into account. A fair comparison of the selected methods is obtained this way, even though the

absolute values of the quality metrics might be different with a more complete ground truth data.

5.3.4 Analysis of Results

In the following, we present the results of the experimental evaluation of above described methods. To the best of our knowledge, such large and comprehensive evaluation of multi-modal retrieval with human-evaluated relevance has not been previously performed. The data we obtained from the experiments thus represent a unique resource for analysis of the fusion methods' behavior.

Aggregation function tuning As discussed earlier, we currently limit our study to two modalities, which express textual and visual similarity of images. To allow multi-modal query processing, we further need to specify how these modalities should be combined. In this section, we briefly comment on the choice and tuning of the aggregation function that was applied in the experiments to facilitate the actual fusion.

Even though late fusion methods principally allow users to define (or at least, adjust) the aggregation function, in our experiments the aggregation needed to be fixed to allow a fair comparison of examined methods. We decided to employ a simple linear combination of the mono-modal distances, which is a straightforward solution that has been successfully applied in many other fusion scenarios (e.g. [57]). Both visual- and text-induced distances were first normalized, and the linear aggregation was tested with several weight settings.

Interestingly, a balanced combination of modalities achieved the best precision of results for both symmetric and asymmetric fusion solutions, even though asymmetric approaches reported in literature typically give more weight to secondary modalities in the postprocessing fusion phase or consider only the secondary modalities for re-ranking. Our results thus agree with the findings of [48] who also recommend to use all available modalities in the postprocessing phase. Therefore, a multi-modal distance function with a balanced combination of both modalities is considered in the postprocessing phase of all following comparisons.

Problem 1: *Efficiency and effectiveness of late-fusion paradigms in large-scale*

In the first set of evaluations, we compare the performance of different late fusion methods on the high-quality Profiset collection. Precise late fu-

sion is represented by the Threshold Algorithm (TA), approximate solutions are the following: approximate TA, re-ranking solutions based on visual (V) modality, inherent fusion with visual primary modality, re-ranking based on text (T) initial search, and inherent fusion with text as the primary modality.

Efficiency The average response times for the monitored fusion techniques can be seen in Figure 5.10a. We can see that the times for the two baseline single-modal searches (visual and text) are increased in the postprocessing phase by approximately 200-300 milliseconds (which constitutes about 30 % increase) in all cases. This represents the time needed to pass the candidate set to the ranking phase and compute the combined distances. Quite noticeable are the high costs of the approximate TA that are about two times higher than in case of postprocessing of the same number of candidate objects. This is caused by the need to access two index structures, which results in increased communication costs. The indices also compete for the single machine resources. This is improved as the parallelization is increased using more CPUs but still the method is nearly two times slower than the asymmetric postprocessing fusion. Out of the scope of the graph is the time of the precise TA that took about 1.5 minutes to compute on average, which is caused mainly by the fact that the ordered lists of candidates needed to be examined very deeply before the precise stopping condition was satisfied.

Distance-based result quality According to the rED(k) measure depicted in Figure 5.10b, the text-based initial search followed by re-ranking of a small $\mathcal{C}^{BS}$ has by far the worst precision of results, which suggests that the top results of text search are not much relevant from the visual perspective. This phenomenon is less pronounced in case of methods that exploit the visual modality as the primary one. With the increasing size of $\mathcal{C}^{BS}$, the retrieval accuracy gradually improves for all approximate techniques. For the comparable size of the candidate set the approximate TA outperforms the asymmetric techniques, since it considers top-ranking objects from both modalities.

User-perceived quality The user opinion on result quality was evaluated by the NDCG metric, which was applied in two modes: in the *natural* mode, objects that were marked as *partially relevant* during the user relevance assessments are considered as having non-zero relevance, whereas in

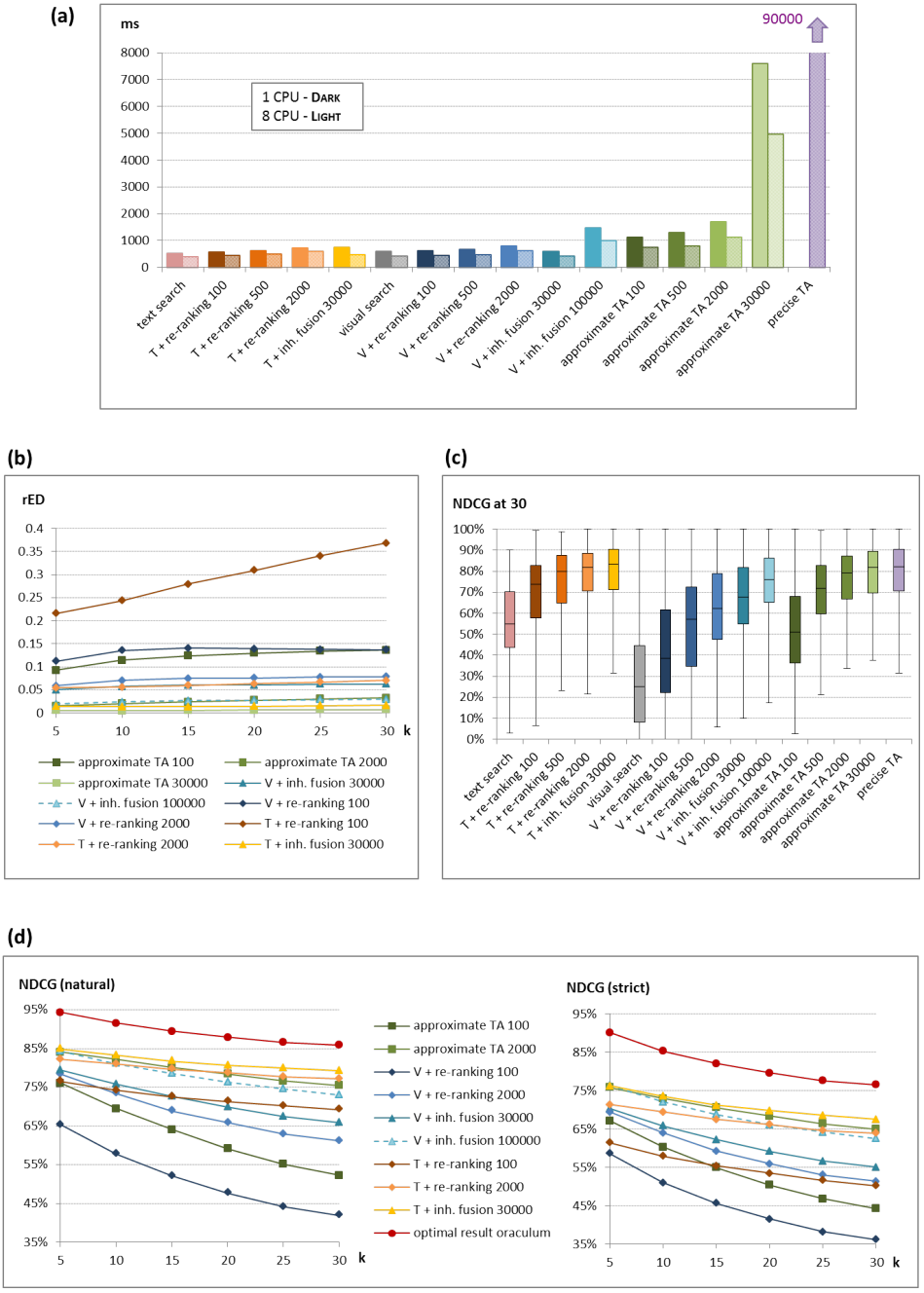

Figure 5.10: Selected late fusion methods performance on Profiset collection: (a) processing costs, (b) average rED at a given rank, (c) quartile distribution of NDCG at 30, (d) average NDCG at a given rank.

the *strict* mode, only objects that were marked as *highly relevant* are deemed relevant. The strict measure thus represents a more demanding user. Figure 5.10c summarizes the user-perceived retrieval quality of all methods we studied, while Figure 5.10d provides a more detailed view on the behavior of methods that are efficient enough (with respect to results shown in Figure 5.10a) to be applicable to interactive large-scale searching (i.e. with response times of a few seconds at most).

In the graphs, we can observe some differences from the distance-based evaluation. The precise TA is no longer dominant, the text-based inherent fusion achieved marginally better results. Moreover, there is a significant difference between the user-perceived quality of text- and visual-based reranking methods. We were able to identify two factors that increase the success of text-based approaches: 1) users tend to prefer semantic relevance, which is typically contained in the text descriptor, over visual similarity; 2) the text modality is more selective – there is a distinct diversification of relevant and irrelevant objects, and the irrelevant cannot enter the postprocessing phase, whereas for visual modality there is no such clear cut.

We can also see that for all asymmetric processing methods, the relevance of results continues to increase with the growing size of $\mathcal{C}^{BS}$ and $\mathcal{C}^{S}$. However, it is important to notice that the dependence between the objective quality and the initial result set size is approximately logarithmic. This is clearly visible for visual-based approaches, which we tested with more values of super-candidate set size $\mathcal{C}^{S}$. For text-based solutions, $\mathcal{C}^{S}$ larger than 30000 would not bring noticeable improvements, as there are not enough objects relevant from the text perspective that could enter the fusion phase. Even for the 30000 limit, about 40 % of our queries did not have that many text candidates.

The graphs in Figure 5.10d contain an additional curve, denoted as *optimal result oraculum*. This line shows the average precision of the best result provided by any of the methods we tested. If we were able to guess which retrieval method is best suited for which query, we could increase the average result relevance by 10 % as compared to the best available method, i.e. the text-based inherent fusion. Obviously, deciding the suitability of a given retrieval method for a given query is a very challenging task and it remains open for future study. We provide some insights into this topic in the discussion of Problem 3.

Finally, let us mention that we also evaluated the relevance of results using the Precision at k metric. The results were very similar to those reported for the NDCG measure both in trends and absolute values of the metric, therefore we not detail them here.

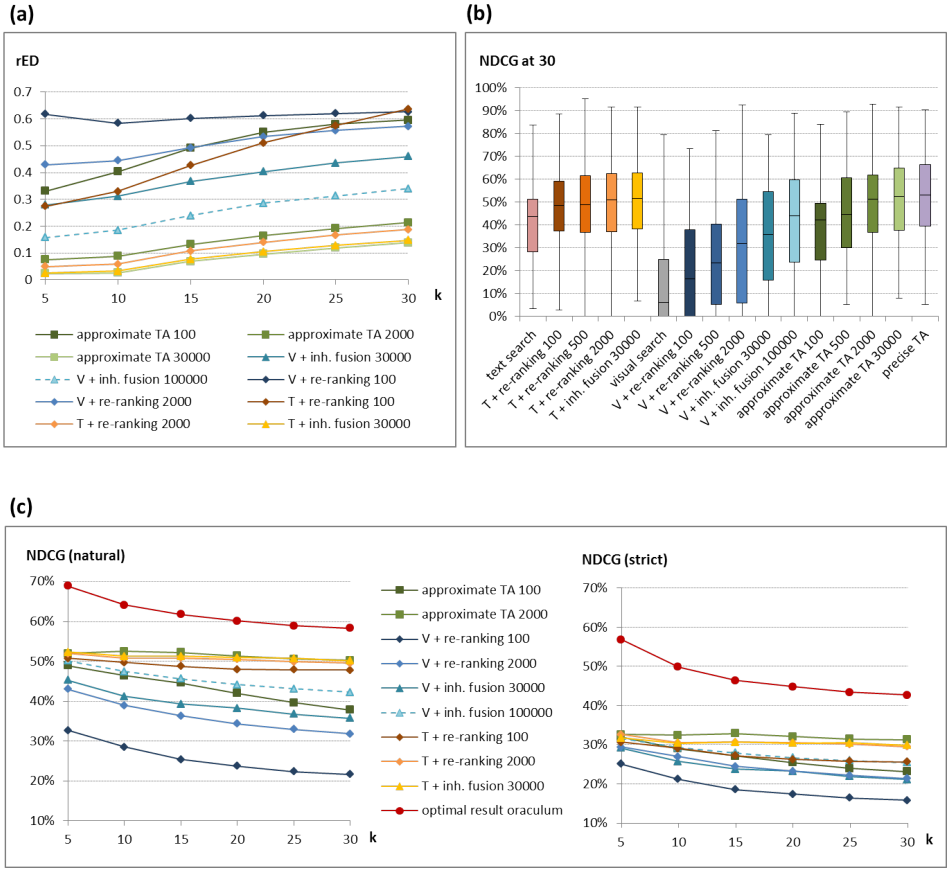

Figure 5.11: Selected late fusion methods performance on CoPhIR collection: (a) average rED at a given rank, (b) quartile distribution of NDCG at 30, (c) average NDCG at a given rank.

Conclusion Taking into consideration both the search effectiveness and efficiency, the text-based retrieval with inherent fusion is the optimal method for the given dataset. Approximate TA and visual-based inherent fusion closely follow in effectiveness but require higher processing times.

Problem 2: *Influence of dataset quality on late fusion techniques performance*

As anticipated, many aspects of the behavior of fusion methods may be related to the properties of the particular dataset employed for testing. To find out which characteristics are dataset-dependent and which are not, we repeated the same set of experiments over the CoPhIR dataset. Fig-

ure 5.11 summarizes the effectiveness results, the costs being approximately the same for both datasets since identical descriptors, distance measures and data volumes were used.

We can see that in general, the result quality trends are quite similar to those we obtained for the Profiset data. The absolute NDCG values are significantly lower but this could be expected since the CoPhIR dataset quality is rather low. We also expected to see that visual-based methods provide better results than text-based, as the text metadata is sparse and erroneous in CoPhIR. However, the experimental results contradict this assumption. Further analysis revealed that the observed behavior is caused by the visual quality of images, which is also significantly lower in the CoPhIR dataset than in the Profiset collection. The Profiset data typically contains professional, "illustrative" images of given concepts, with no disturbing elements. The CoPhIR collection, on the other hand, is composed of amateur photos, which are often of low quality in terms of both image capturing (focus, noise, ...) and composition (too many objects, complex background, etc.). The similarity between the low-quality images is more difficult to evaluate for both people (as observed during the relevance assessments) and the MPEG-7 visual descriptors. It should also be noted that the average number of relevant images per query, found together by all evaluated methods, is lower for the CoPhIR dataset than for Profiset (89 perfect objects vs. 169, respectively). On one hand, this may illustrate the inefficiency of the retrieval methods, on the other hand, it is likely that the CoPhIR dataset contains many more "useless" images than Profiset and it is thus more challenging to identify the few relevant ones.

Conclusion The most important implications from the relevance evaluations over both Profiset and CoPhIR collections are the following:

- Multi-modal methods significantly outperform the mono-modal ones in terms of result relevance.
- Efficient approximate fusion techniques are able to provide results of comparable quality to the precise Threshold Algorithm solution, while greatly reducing the evaluation costs.
- If we consider the average result relevance for an arbitrary query, asymmetric fusion techniques with content-based basic search are the least successful from the proposed multi-modal solutions. For collections with high-quality text metadata, text-based asymmetric fusion seems to be the most suitable. For worse datasets or datasets with

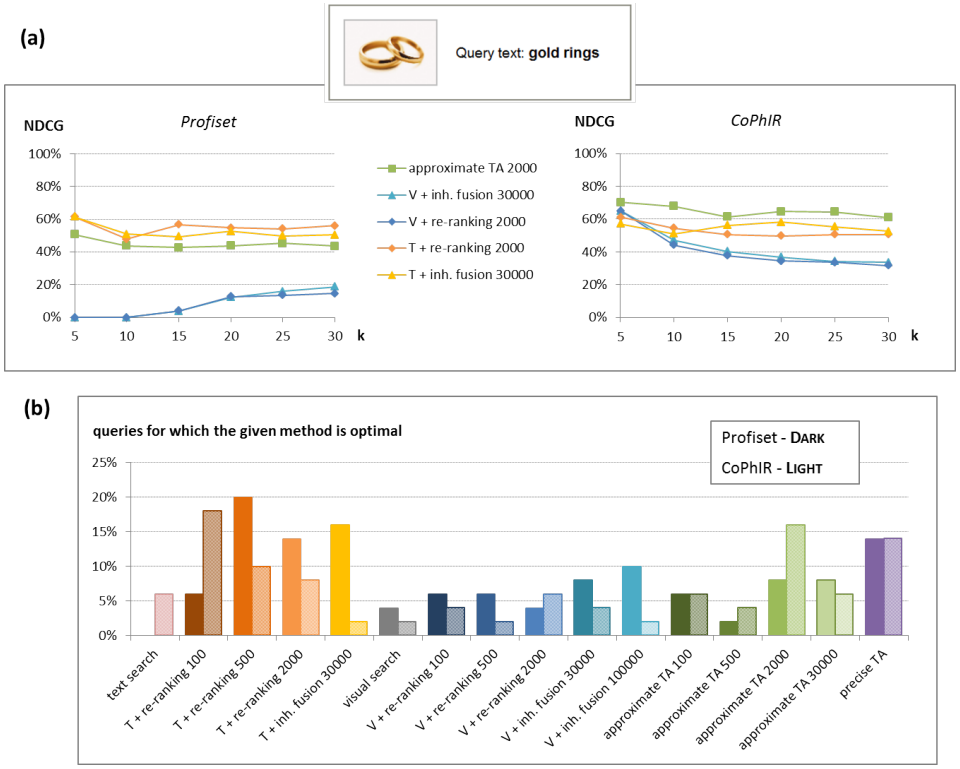

Figure 5.12: (a) Performance of selected methods for a specific query in different datasets, (b) percentage of queries for which the given methods were optimal in the respective datasets.

unknown quality, the symmetric fusion can be recommended as it is less affected by a potential ineffectiveness of one modality.

Problem 3: *Query-level analysis of retrieval performance*

Apart from the averaged results that estimate the suitability of given methods for an arbitrary query, it is also interesting to look more thoroughly on the individual query objects in our selection, and try to identify some rules that determine the effectiveness of a given search methods for a given query. In this section, we focus on two aspects: 1) a high-level analysis of trends that can be observed in our experimental results, and 2) possibilities of automatic selection of a suitable search method using the distance properties of the result set.

On the high analytical level, we are interested in the following characteristics: a comparison of average performance of a given method to performances for particular objects, and a comparison of relevance achieved by a particular method for a particular query in the two different datasets. Analysis of results, some outputs of which are visualized in Figure 5.12, reveals the following:

- The behavior of individual queries largely varies from the average case for both test datasets. As depicted in Figure 5.12b, a mono-modal text retrieval in Profiset collection is the only approach that never provides an optimal result. The overall-best approaches (text-based inherent fusion for Profiset, approximate TA for CoPhIR) are actually optimal for only about 16 % of results in both experiments. However, text-based approaches remain dominant, followed by the symmetric TA solution.

- The behavior of the same method for the same query also frequently differs between the two datasets, which can be seen in general in Figure 5.12b and also in a particular example depicted in Figure 5.12a. This shows us that the suitability of a given search method depends on the characteristics of the dataset as well as the query object itself.

The above two observations clearly show that the multi-modal image retrieval is a very challenging task. There are no clear distinctions between suitable and unsuitable methods, and there are only a few trends that can be considered universal across datasets. We can manually identify some semantic features that are in relation with the performance of specific search methods, e.g. the visual-based approaches are more suitable for queries with a rather uniform visual representation (“water”,“sunset”) and can also perform well for ambiguous or broad queries (“shells”, “stamp”,“bird”) if enough visually similar objects exist in a given dataset. However, it is very difficult to put these observations into relation with some objectively measurable properties of the queries, datasets, or results.

Since the data we gathered in the experiments contains also a lot of information about distances between query objects and results, we actually tried to look for correlations or rank dependencies between the query method performance and the mean and the standard deviation values of the distances in the respective result. This analysis assumed that results with low distances to the query object are likely to be relevant (this is however not true vice-versa, as higher distances may also imply a more complex query) and that good results contain objects similar to each other (which

	T + inh. fusion 30000 [NDCG(30)]	**V + inh. fusion 30000 [NDCG(30)]**	**approx. TA 2000 [NDCG(30)]**
avg. dist. [P]	-0.37 (r), -0.26 (τ)	-0.46 (r), -0.32 (τ)	-0.24 (r), 0.16 (τ)
avg. dist. [C]	-0.37 (r), -0.20 (τ)	-0.47 (r), -0.33 (τ)	-0.29 (r), -0.21 (τ)
std. dev. [P]	0.10 (r), -0.11 (τ)	-0.22 (r), -0.18 (τ)	-0.20 (r), 0.02 (τ)
std. dev. [C]	-0.07 (r), -0.10 (τ)	-0.18 (r), -0.22 (τ)	0.31 (r), 0.24 (τ)

Table 5.4: Pair-wise correlations between the relevance of selected methods and descriptive characteristics of the respective search results for Profiset (P) and CoPhIR (C) datasets. The correlations were measured by the Pearson correlation coefficient r and Kendall's tau coefficient τ.

would be reflected in the low standard deviation). We evaluated the dependencies between these characteristics and the NDCG at 30 of each method – several results can be seen in Table 5.4. Furthermore, we tested whether some dependencies exist between a difference in NDCG and a difference in descriptive characteristics for selected pairs of methods. Unfortunately, none of these analyses revealed significant relationships.

Conclusion The results discussed in this section show that the performance of a particular search method for a given query and a given dataset is influenced by many factors and may be significantly different from the average case. Our first lightweight analysis of distance distributions in result sets has not provided any reliable method of deciding which approach should be used for a given query. To be able to study this problem more thoroughly, it will be necessary to create a detailed model of measurable properties of query processing, collect such data and submit them to a thorough statistical evaluation. It would also be helpful if more objects could be included in the experimental evaluation, this is however limited by the need to collect the relevance assessments, which is a tedious process.

Problem 4: *Usefulness of query expansion for large-scale image search*

As promised in Section 5.3.1, we performed another set of experiments with the same set of queries and search methods, but this time the query keywords were expanded by keywords from manually selected WordNet synsets. The objective of this experiment was to discover what result relevance improvements can be achieved by expansion when relevant expan-

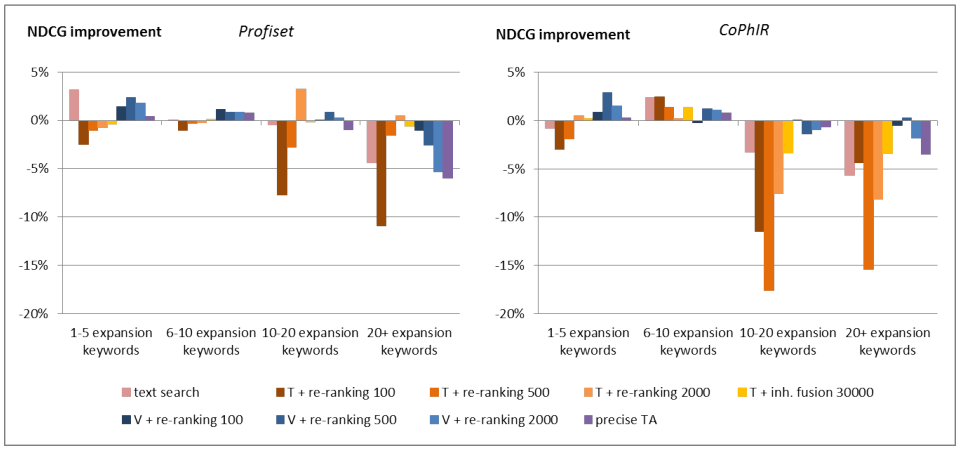

Figure 5.13: Effects of query expansion on result relevance.

sion inputs are available. This should help us to decide whether it is worth developing automatic expansion methods. To prevent the number of results to be evaluated by human assessors from becoming too high, only selected methods were employed for the expanded queries.

As with the previous problems, we not only evaluated the overall impact of query expansion on the quality of results, but also tried to identify some factors that influence the usefulness of expansion. On the overall level, the expansion did not meet our expectations, as the average relevance values remained generally the same. The relevance improvement for individual queries, measured by the change of NDCG(30) score, followed the normal distribution, with the mean improvement value ranging from approximately -3 % to 1 % in both datasets. The best improvement was observed for visual-based search with multi-modal re-ranking of top 500 candidate objects, the worst results were recorded for text-based search with re-ranking of top 100 candidates.

We also studied the dependency of relevance improvement on the following two characteristics: 1) number of original query keywords, and 2) number of keywords added by the expansion. In the first case, no interesting dependency was found, but the second comparison revealed some relationship. As depicted in Figure 5.13, the expansion is more successful when less than 10 new keywords are added than in the other cases. We can also see that visual-based asymmetric techniques most frequently profit on the expansion, whereas for text-based approaches the risk of introducing noise is rather high.

Conclusion The authors of an expansion survey study [39] state that using WordNet for query expansion is advantageous only if the query words are disambiguated almost exactly. In our experiment, we did provide exact disambiguation manually, however the advantages of expansion are unconvincing. Therefore, we believe that developing WordNet-based expansion for large-scale image retrieval is not a promising approach for future research.

5.4 Query Language for Complex Similarity Queries

In the survey of search techniques in Chapters 3 and 4 we could see that a lot of intensive research has been conducted recently in the field of indexing methods and search algorithms for similarity-based retrieval. As a result, state-of-the-art search systems already support quite complex similarity queries with a number of features that can be adjusted according to individual user's preferences. To communicate with such a system, it is either possible to employ low-level programming tools, or a higher-level communication interface that shields users from the implementation details employed by the particular search engine. As the low-level tools can only be used by a limited number of specialists, the high-level interface becomes a necessity when common users shall be allowed to issue advanced queries or adjust the parameters of the retrieval process. To address this need, we propose such high-level interface in a form of a structured query language that allows users to issue advanced queries over complex data.

The motivation to study query languages arose also from the development of the MUFIN search system. The system and the underlying MESSIF library currently offer a wide spectrum of retrieval algorithms that are used to support several multimedia search applications, such as large-scale image search, automatic image annotation, or gait recognition. So far, users are allowed only to select the query object via a graphical interface, and the choice of the actual search methods as well as its parameters and other settings are hard-coded into the system. To improve the usability of our systems, we decided to offer a query language that would allow advanced users to express their preferences without having to deal with the technical details. After a thorough study of existing solutions we came to a conclusion that none of them covers all our specific needs. Therefore, we decided to propose a new language based on and extending the existing ones. At the same time, it was our desire to design the language in such a way that it could also be used by other systems.

Consequently, we present an SQL-based query language which can be used to formulate a wide range of similarity queries, as we demonstrate on examples from various application domains. Building on a thorough analysis of previous studies and our long-time experience with both theory and practice of similarity search systems, we have proposed its structure so that it supports all fundamental query types and can be easily extended. The language can be used by programmers or advanced users to issue queries in a standard declarative way, shielding them from the execution details. For less advanced users, we expect the language to be wrapped-up into a visual interface. The language is designed in a general way to be flexible and extensible.

5.4.1 Available Languages for Multimedia Retrieval

The problem of defining a formal apparatus for similarity queries has been recognized and studied by the data processing community for more than two decades, with various research groups working on different aspects of the problem. Some of these studies focus on the underlying algebra, others deal with the query language syntax. Query languages can be further classified as SQL-based, XML-based, and others with a less common syntax. In the following, we briefly survey the main directions of query language development that influenced the design of our language.

The majority of early proposals for practical query languages are based on SQL or its object-oriented alternative, OQL. Paper [100] describes MOQL, a multimedia query language based on OQL which supports spatial, temporal, and containment predicates for searching in image or video. However, similarity-based searching is unsupported in MOQL. In [73], a more flexible similarity operator for nearest neighbors is provided but its similarity measure cannot be chosen. Commercial products, such as Oracle or IBM DB2, follow the strategy outlined in the SQL/MM standard [112], which recommends to incorporate the similarity-based retrieval into SQL via user-defined data types and functions.

Much more mature extensions of relational DBMS and SQL are presented in [13, 77]. The concept of [13] enables to integrate similarity queries into SQL, using new data types with associated similarity measures and extended functionality of the select command. The authors also describe the processing of such extended SQL and discuss optimization issues. Even though the proposed SQL extension is less flexible than we need, the presented concept is sound and elaborate. The study [77] only deals with image retrieval but presents an extension of the PostgreSQL database man-

agement system that also allows to define feature extractors, create access methods and query objects by similarity. This solution is less complex than the previous one but, on the other hand, it allows users to adjust the weights of individual features for the evaluation of similarity.

Recently, we could also witness interest in XML-based languages for similarity searching. In particular, the MPEG committee has initiated a call for proposal for MPEG Query Format (MPQF). The objective is to enable easier and interoperable access to multimedia data across search engines and repositories [62]. The format supports various query types (by example, by keywords, etc.), spatio-temporal queries and queries based on user preferences. From among various proposals we may highlight [154] which presents an MPEG-7 query language that also allows to query ontologies described in OWL syntax.

5.4.2 Query Language Design

As discussed above, we desire to create a query language that can be used to define advanced queries over multimedia or other complex data types. The language should be general and extensible, so that it can be employed with various search systems. To achieve this, we first analyzed the desired functionality of the language. Subsequently, fundamental design decisions concerning the language architecture were taken.

Analysis of Requirements

As detailed in [35], we thoroughly studied the following three sources to collect requirements for a multimedia query language: (1) the current trends in multimedia information retrieval, which reveal advanced features that should be supported by the language; (2) existing query languages and their philosophies, so that we can profit on previous work; and (3) the MESSIF framework architecture. The following issues were identified as the most important:

- support for a wide range of query types: in addition to various search algorithms, such as nearest neighbor search, range queries, similarity joins, sub-sequence matching, etc., single- and multi-object similarity queries as well as attribute-based (relational) and spatio-temporal queries need to be taken into consideration;
- support for multi-modal searching: multiple information sources and complex queries that allow to combine attribute-based, text-based and

similarity-based search are a fundamental part of modern information retrieval;

- adjustability of searching: users need means of expressing their preferences in various parameter settings (e.g. precise vs. approximate search, user-defined distance functions, or distance aggregation functions);
- support for query optimization: optimizations are vital for efficient evaluation of complex queries in large-scale applications.

Language Fundamentals

The desired functionality of the new language comprehends the support for standard attribute-based searching which, while not being fully sufficient anymore, still remains one of the basic methods of data retrieval. A natural approach to creating a more powerful language therefore lies in extending some of the existing, well-established tools for query formulation, provided that the added functionality can be nested into it. Two advantages are gained this way: only the extended functionality needs to be defined and implemented, and the users are not forced to learn a new syntax and semantics.

The two most frequently used formalisms for attribute data querying are the relational data model with the SQL language, and the XML-based data modeling and retrieval. As we could observe in the related work, both these solutions have already been employed for multimedia searching, but they differ in their suitability for various use cases. The XML-based languages are well-suited for inter-system communication while the SQL language is more user-friendly since its query structure imitates English sentences. In addition, SQL is backed by a strong theoretical background of relational algebra, which is not in conflict with content-based data retrieval. Therefore, we decided to base our approach on the SQL language, similar to existing proposals [13, 77].

By employing the standard SQL [144] we readily gain a very complex set of functions for attribute-based retrieval but no support for similarity-based searching. Since we aim at providing a wide and extensible selection of similarity queries, it is also not possible to employ any of the existing extensions to SQL, which focus only on a few most common query operations. Therefore, we created a new enrichment of both the relational data model and the SQL syntax so that it can encompass the general content-based retrieval as discussed above.

The reasons for introducing new language primitives instead of utilizing user-defined functions are discussed in [13]. Basically, treating the content-based operations as "first-class citizens" of the language provides better opportunities for optimizations of the query evaluation. In our solution, we follow the philosophy of [13] but provide a generalized model for the content-based retrieval.

System Architecture

In the existing proposals for multimedia query languages based on SQL, it is always supposed that the implementing system architecture is based on RDBMS, either directly as in [77], or with the aid of a "blade" interface that filters out and processes the content-based operations while passing the regular queries to the backing database [13].

Both these solutions are valid for our new query language. Since we propose to extend the SQL language by adding new language constructs, these can be easily intercepted by a "blade", evaluated by an external similarity search system, and passed back to the database where the final results are obtained. The integration into a RDBMS follows an inverse approach. The database SQL parser is updated to support the new language constructs and the similarity query is evaluated by internal operators.

One of our priorities is creating a user-friendly tool for the MESSIF library. The storage backend of the MESSIF utilizes a relational database and the functionality of the standard SQL is thus internally supported. Therefore, we only need to provide a parser of the query language and a translation to native MESSIF API calls and let the framework take care of the actual execution.

5.4.3 Data Model and Operations

The core of any information management system is formed by data structures that store the information, and operations that allow to access and change it. To provide support for content-based retrieval, we need to revisit the data model employed by SQL and adjust it to the needs of complex data management.

Data Model

On the concept level, multimedia objects can be analyzed using standard entity-relationship (ER) modeling. In the ER terminology, a real-world ob-

ject is represented by an *entity*, which is formed by a set of descriptive object properties – *attributes*. The attributes need to contain all information required by target applications. In contrast to common data types used in ER modeling, which comprise mainly text and numbers, attributes describing multimedia objects are often of more complex types, such as image or sound data, time series, etc. The actual attribute values form an *n-tuple* and a set of n-tuples of the same type constitute a *relation*.

Relations and attributes (as we shall continue to call the elements of n-tuples) are the basic building blocks of the Codd's relational data model and algebra [49], upon which the SQL language is based. This model can also be employed for complex data but we need to introduce some extensions. A relation is defined as a subset of the Cartesian product of sets $\mathcal{D}_1$ to $\mathcal{D}_n$, $\mathcal{D}_i$ being the domain of attribute A_i. Standard operations over relations (selection, projection, etc.) are defined in first-order predicate logic and can be readily applied on any data, provided the predicates can be reasonably evaluated. To control this, we use the concept of *data type* that encapsulates both a specification of an attribute domain and the functions that can be applied on members of this domain. Let us note here that Codd used a similar concept of *extended data type* in [49], but he only worked with a few special properties of the data type, in particular the total ordering. As we shall discuss presently, our approach is more general. We allow for an infinite number of data types, which may directly represent objects (e.g. image, sound) or some derived information (e.g. color histogram). The derived data characteristics correspond to modalities and their values are obtained from the multimedia objects by projection functions of the respective modalities.

According to the best-practices of data modeling [144], redundant data should not be present in the relations, which also concerns derived attributes. To minimize the space complexity and avoid the data inconsistency threat, derived attributes should only be computed when needed in the process of data management. In case of complex data, however, the computation of the projection can be very costly. Thus, it is more suitable to allow storing some derived attributes in relations, especially when these are used for data indexing. Naturally, more projection functions may be available to derive additional attributes when asked for. Figure 5.14 depicts a possible representation of an image object in a relation.

Operations on Data Types

As we already stated, each data type consists of a specification of a domain of values, and a listing of available functions. As some of the functions

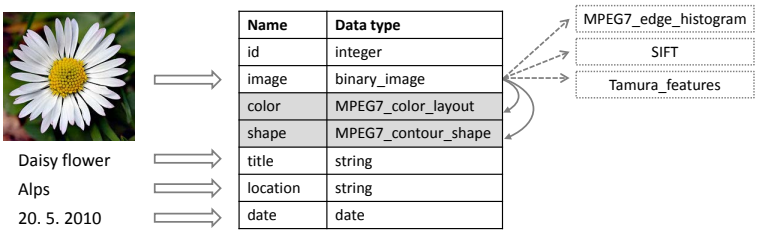

Figure 5.14: Transformation of an image object into n-tuple. Full and dashed arrows on the right side depict materialized and available projection functions, respectively.

are vital for the formulation and execution of the algebra operations, we introduce several special classes of functions that may be associated with each data type:

- *Comparison functions*: Functions of this type define total ordering of a given domain ($c : \mathcal{D} \times \mathcal{D} \rightarrow \{<, =, >\}$). When a comparison function is available, standard indexing methods such as B-trees can be applied and queries using value comparison can be evaluated. Comparison functions are typically not available for multimedia data types and the data types derived from them, where no meaningful ordering of values can be defined.

- *Distance functions*: Distance functions evaluate the dissimilarity between two values from a given data domain ($d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_0^+$). We do not impose any restrictions on the behavior of d in general, but there exists a way of registering special properties of individual functions that will be discussed later. More than one distance function can be assigned to a data type, in that case one of the functions needs to be denoted as default. When more distance functions are available for a given data type, the preferred distance function can be specified in a relation definition. In case no distance function is provided, a trivial *identity distance* is associated to the data type, which assigns distance 0 to a pair of identical values and distance ∞ to any other input.

- *Projection functions*: Projection functions transform values of one data type into the values of a different data type ($p : \mathcal{D}_i \rightarrow \mathcal{D}_j$). Projections are typically used on complex unstructured data types (such as binary image) to produce data types more suitable for indexing and retrieval

(e.g. color descriptor). An arbitrary number of projection functions can be associated to each data type.

In addition to the declaration of functionality, each of the mentioned operations can be equipped with a specification of various properties. The list of properties that are considered worthwhile is inherent to a particular retrieval system and depends on the data management tools employed. For instance, many indexing and retrieval techniques for similarity searching rely on certain properties of distance functions, such as the metric postulates or monotonicity. To be able to use such a technique, the system needs to ascertain that the distance function under consideration satisfies these requirements. To solve this type of inquiries in general, the set of properties that may influence the query processing is defined, and the individual functions can provide values for those properties that are relevant for the particular function.

Operations on Relations

The functionality of a search system is provided by operations that can be evaluated over relations. In addition to standard selection and join operations, multimedia search engines need to support various types of similarity-based retrieval. Due to the diversity of possible approaches to searching, we do not introduce a fixed set of operations but expect each system to maintain its own list of methods. Each operation needs to specify its input, which consists of 1) number of input relations (one for simple queries, multiple for joins), 2) expected query objects (zero, singleton, or arbitrary set), and 3) operation-specific parameters, which may typically contain a specification of a distance function, distance threshold, or operation execution parameters such as approximation settings. Apart from a special case discussed later the operations return relations, typically with the schema of the input relation or a Cartesian product of input relations. In case of similarity-based operations the schema is enriched with additional *distance* attribute which carries the information about the actual distance of a given result object with respect to the query distance function.

Similar to operations on data types, operations on relations may also exhibit special properties that can be utilized by the search engine. In this case, the properties are mostly related to query optimization. As debated earlier, it is not possible to define general optimization rules for a model with a variable set of operations. However, each search system can maintain its own set of optimization rules together with the list of operations.

A special subset of operations on relations is formed by functions that produce scalar values. Among these, the most important are the *generalized distance* functions that operate on relations and return a single number, representing the distance of objects described by n-tuples. The input of these functions contains 1) a relation representing the object for which the distance needs to be evaluated, 2) a relation with one or more query objects, and 3) additional parameters when needed. A typical example of the generalized distance function is a set distance that utilizes multiple query objects, or an aggregated distance that combines several partial distances into overall similarity evaluation.

Data Indexing

While not directly related to the data model, data indexing methods are a crucial component of a retrieval system. The applicability of individual indexing techniques is limited by the properties of the target data. To be able to control the data-index compatibility or automatically choose a suitable index, the search system needs to maintain a list of available indices and their properties. The properties can then be verified against the definition of the given data type or distance function (basic or generalized). Thus, metric index structures for similarity-based retrieval can only be made available for data with metric distance functions, whereas traditional B-trees may be utilized for data domains with total ordering. It is also necessary to specify which search operations can be supported by a given index, as different data processing is needed e.g. for the nearest-neighbor and reverse-nearest-neighbor queries. Apart from the specialized indices, any search system inherently provides the basic *Sequential Scan* algorithm as a default data access method that can support any search operation.

5.4.4 SimSeQL

Having defined the underlying data model, we now proceed with the specification of the syntax and semantics of the new language. We also provide several examples that illustrate the use of the novel language constructs.

Syntax and Semantics

The SimSeQL language is designed to provide a user-friendly interface to state-of-the-art multimedia search systems. Its main contribution lies in enriching the SQL by new language constructs that enable to issue all kinds

of content-based queries in a standardized manner. In accordance with the declarative paradigm of the SQL, the new constructs allow to describe the desired results while shielding users from the execution issues. On the syntactical level, the SimSeQL contributes mainly to the query formulation tools of the SQL language. Data modification and control commands are not discussed here but their adaptation to the generalized data types and operations is straightforward. On the semantic level, however, the original SQL is significantly enriched by the introduction of an unlimited set of complex data types and related operations.

A SimSeQL query statement follows the same structure as SQL, being composed of the six basic clauses SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY, with their traditional semantics [144]. The extended functionality is mainly provided by a new construct SIMSEARCH, which is embedded into the FROM clause and allows to search by similarity, combine multiple sources of information, and reflect user preferences. Prior to a detailed description of the new primitives, we present the overall syntax in the following schema:

```
SELECT      [TOP n | ALL]
            {attribute | ds.distance | ds.rank | f(params)} [, ...]
FROM        {dataset |
            SIMSEARCH [:obj [, ...]]
                IN data_source AS ds [, data_source2 [, ...]]
                BY {attribute [DISTANCE FUNCTION
                                    distance_function(params)]
                    | distance_function(params)}
                [METHOD method(params)]
WHERE       /* restrictions of attribute values */
ORDER BY    {attribute | ds.distance [, ...]}
```

In general, there are two possible approaches to incorporating primitives for content-based retrieval into the SQL syntax. We can either make the similarity search results form a new information resource on the level of other data collections in the FROM clause (an approach used in [77]), or handle the similarity as another of the conditions applied on candidate objects in the WHERE clause (exercised in [4, 13, 73, 100]). However, the latter approach requires standardized predicates for various types of similarity queries, their parameters etc., which is difficult to achieve in case an extensible set of search operations and algorithms is to be supported. In addition,

the similarity predicates are of a different nature than attribute-based predicates and their efficient evaluation requires specialized data structures. Therefore, we prefer to handle similarity-based retrieval as an independent information source. Consequently, we only standardize the basic structure and expected output, which can be implemented by any number of search methods of the particular search engine.

As anticipated, the similarity-based retrieval is wrapped in the SIMSEARCH language construct, which produces a standard relation and can be seamlessly integrated into the FROM clause. The SIMSEARCH expression is composed of several parts explained in the following sections.

Specification of query objects The selection of query objects follows immediately after the SIMSEARCH keyword. An arbitrary number of query objects can be provided, each object being considered an attribute that can be compared to attributes of the target relations. Multiple query objects can be used to express a more complex information need. A query object (attribute) can be represented directly by an attribute value, by a reference to an object provided externally, or by a nested query that returns the query object(s). The query objects need to be type-compatible with the attributes of the target relation they are to be compared to.

Specification of a target relation The keyword IN introduces the specification of one or more relations, elements of which are processed by the search algorithm. Each relation can be produced by a nested query.

Specification of a distance function An essential part of a content-based query is the specification of a distance function. The BY subclause offers three ways of defining the distance: calling a distance function associated to an attribute, referring directly to a distance function provided by the search engine, or constructing the function within the query. In the first case, it is sufficient to enter the name of attribute to invoke its default distance function. Non-default distance function of an attribute needs to be selected via the DISTANCE FUNCTION primitive that also allows to pass additional parameters for the distance function if necessary. The last case allows greater freedom of specifying the distance function by the user, but both the attributes for which the distance is to be measured must be specified. A special function DISTANCE(x, y) can be used to call the default distance function defined for the given data type of attributes x, y. The nuances of referring to a distance function can be observed in the following:

```
SIMSEARCH ... BY color
  /* search by the default distance function of the color attribute */

SIMSEARCH ... BY color DISTANCE FUNCTION color_distance
  /* search by color_distance function of the color attribute */

SIMSEARCH ... BY some_special_distance(qo, color, param)
  /* search by some_special_distance applied to the query
     object qo, color attribute, and an additional parameter */

SIMSEARCH ... BY DISTANCE(qoc, color)+DISTANCE(qos, shape)
  /* search by a user-defined sum of the default distance functions
     on qoc and qos query objects and color and shape attributes */
```

Specification of a search method The final part of the SIMSEARCH construct specifies the search method or, in other words, the query type (e.g. range query, similarity join, distinct nearest neighbor search, etc.). Users may choose from a list of methods offered by the search system. It can be reasonably expected that every system supports the basic nearest neighbor query, therefore this is considered a default method in case none is specified with the METHOD keyword. The default nearest neighbor search returns all n-tuples from the target relation unless the number of nearest neighbors is specified in the SELECT clause by the TOP keyword.

The complete SIMSEARCH phrase returns a relation with a schema of the target relation specified by the IN keyword, or the Cartesian product in case of more source relations. Moreover, information about distance of each n-tuple of the result set computed during the content-based retrieval is available. This can be used in other clauses of the query, referenced either as *distance*, when only one distance evaluation was employed, or prefixed with the named data source in the clause if ambiguity should arise.

Example Scenarios

To illustrate the wide applicability of the SimSeQL language, we now present several query examples for various use-case scenarios found in image and video retrieval (more can be found in [35]). Each of the use-cases is accompanied by a short comment on the interesting language features employed. For the examples, let us suppose that the following relations, data types and functions are available in the retrieval system:

- **video_frame** relation: list of video frames

name	**type**	**distance function**
id	integer	identity_distance *(default)*
video_id	integer	identity_distance *(default)*
video	binary_video	identity_distance *(default)*
face_descriptor	number_vector	mpeg7_face_metric *(default)*
subtitles	string	tf_idf *(default)*
time_second	long	L1_metric *(default)*

- **image** relation: register of images

name	**type**	**distance function**
id	integer	identity_distance *(default)*
image	binary_image	identity_distance *(default)*
color	number_vector	mpeg7_color_layout_metric *(default)* L1_metric
shape	number_vector	mpeg7_contour_shape_metric *(default)* L2_metric
title	string	tf_idf *(default)*

Query 1 *Retrieve 30 most similar images to a given example*

```
SELECT  TOP 30 id, distance
FROM    SIMSEARCH :queryImage IN image BY shape
```

This example presents the simplest possible similarity query. It employs the default nearest neighbor operation over the shape descriptor with its default distance function. User does not need any knowledge about the operations employed, only selects the means of similarity evaluation. The supplied parameter $queryImage$ represents the MPEG7_contour_shape descriptor of an external query image (provided by a surrounding application). The output of the search is the list of identifiers of the most similar images with their respective distances.

Query 2 *Retrieve images most similar to a set of examples (e.g. identifying a flower by supplying several photos)*

```
SELECT  TOP 1 title
FROM    SIMSEARCH
                extract_MPEG7_color_layout(:o1) AS co1,
                extract_MPEG7_color_layout(:o2) AS co2,
                extract_MPEG7_contour_shape(:o3) AS sh3
            IN image
            BY minimum(DISTANCE(co1, color),
                DISTANCE(co2, color), DISTANCE(sh3,shape))
```

This query represents an example of a multi-object query, input of which are external binary images (denoted as *o1*, *o2*, *o3*) that are transformed to the required descriptors via projection functions. Alternatively, the query objects could be provided as a result of a nested query. The minimum aggregation function employed for similarity evaluation is applied on the distances to individual objects, which are internally linked to the individual attributes and distance functions. Note that the default distance functions of the respective attributes are applied using DISTANCE(x, y) construct.

Query 3 *Retrieve all videos where the Vesuv mountain appears (image similarity) and a commentator mentions volcanoes (speech/text similarity) within two minutes (time aggregation)*

```
SELECT  vf1.video_id
FROM    SIMSEARCH IN
            SIMSEARCH
                extract_MPEG7_color_layout(:VesuvImage) AS co,
                extract_MPEG7_contour_shape(:VesuvImage) AS sh
            IN video_frame AS vf1
                BY weight_sum((DISTANCE(shape,sh), 0.7),
                              (DISTANCE(color, co), 0.2))
                METHOD MessifRangeQuery(0.1,15000)
            NATURAL JOIN
            SIMSEARCH 'vulcano' IN video_frame AS vf2
                BY subtitles
                METHOD MessifRangeQuery(0.1,15000)
        BY DISTANCE(vf1.time_second, vf2.time_second)
        AS sim_frames
WHERE   sim_frames.distance <= 120
```

In this example, multiple modalities are combined to produce the result. In addition, the user selected a special search method that enables to set approximation in work (the second parameter of the search method is the maximum number of objects that may be visited in query processing).

5.5 Summary

In Chapter 5 we have presented our achievements in the field of multi-modal image retrieval, which include a theoretical analysis and modelling of the retrieval process as well as a practical implementation and evaluation of multiple search techniques. Let us now briefly summarize the main contributions contained in this chapter.

In Section 5.2, we have discussed several approximate modality fusion solutions. First, we have presented an extension of the MUFIN search system that provides support for symmetric multi-modal searching, based on the Threshold Algorithm. Apart from introducing a suitable architecture that supports the necessary operations, we have studied the possibilities of approximate searching in the distributed environment, providing users with an estimation of the quality of the approximate result. This work was published in [18]. In Sections 5.2.2 and 5.2.3, we have focused our attention on asymmetric fusion techniques. We have proposed, implemented and evaluated several methods for text-based re-ranking of content-based search results and compared a precise and approximate implementation of the relevance feedback principle. The proposed re-ranking methods were published in [29, 33]. To improve the usefulness of the asymmetric fusion, we have further proposed the inherent fusion technique that allows to implement asymmetric fusion in an efficient and scalable way. This technique was presented in [28].

A more complete view of possible approaches to multi-modal image retrieval has been provided in Section 5.3, which describes our comprehensive evaluation of fusion methods. Multiple late-fusion techniques that can be used to combine textual and visual image similarity have been implemented and compared in terms of both effectiveness and efficiency, using two large-scale datasets and user-provided relevance assessments. The results of this experimental evaluation have been analyzed to answer a set of questions related to the usefulness of different approaches. The methodology and early results of this evaluation were presented in [34], a more thorough analysis of the performance and effectiveness of late-fusion techniques over the Profiset collection is contained in [28]. A comprehensive

publication that will present the whole comparative evaluation is currently being prepared.

Finally, in Section 5.4 we have introduced the SimSeQL, an extensible query language for searching in complex data domains. This language is backed by a general model of data structures and operations, which is applicable to a wide range of search systems that offer different types of content-based functionality. Moreover, the support for data indexing and query optimization is inherently contained in the model. The SimSeQL language was presented in [36], more details can be found in a technical report [35].

Chapter 6

Evaluation in Large-Scale Image Retrieval

In the previous chapter, we have studied the behavior of different retrieval methods in large-scale environments. As discussed earlier, the trade-off between the query processing costs and the user satisfaction can only be measured by experiments over large datasets. However, even the preparation of such evaluation dataset is a challenging task since it is necessary to collect a lot of data, especially the relevance assessments of query-result object pairs. This chapter presents the evaluation platform we created and used in the comparative evaluation, as the platform itself was prepared in an innovative way and provides a contribution to the research field.

The chapter is structured as follows: first, we detail the problematic issues related to image search benchmarking, present existing solutions, analyze their limitations, and explain why none of the existing benchmarks was satisfactory for our experiments. Next, we introduce the new Profiset evaluation platform, focusing in particular on the process of collecting the ground truth data. We discuss the architecture of our solution that allows other research groups to reuse and extend our test data. We also report some interesting statistics that provide valuable insights into human understanding of image similarity.

6.1 Benchmarking Problem

As observed in numerous studies on multimedia retrieval [55, 94, 97, 146], the existence of common benchmarking procedures is essential for evaluation and development of search techniques. To be able to compare different approaches in an objective way, it is necessary to establish a benchmarking platform that allows repeated evaluation of search tasks under fixed conditions, in contrast to various one-time comparisons of approaches reported in many research papers. A complete evaluation test-bed should contain a collection of documents, a set of benchmark queries, a set of ground-truth relevance scores for the benchmark queries, and a set of evaluation met-

rics [94]. The first three components are closely interconnected, whereas the evaluation metrics are independent of a particular data collection and can be studied separately.

Similar to retrieval methods, the evaluation test-bed also needs to be constructed with respect to target applications of the retrieval, reflecting the users' information needs. Accordingly, specialized test collections exist for several narrow domains, e.g. forensic images [75] or coins [122]. However, the situation is more complicated in case of large-scale, broad-domain image retrieval. The major challenge here lies in the acquisition of a large and representative dataset and the necessary ground truth data. Even though several test collections have been created in recent years, none of them has yet been accepted as a common benchmark. Devising methods of collecting test data thus remains a highly desirable topic. On the other hand, suitable evaluation metrics for large-scale retrieval already exist [90, 109, 127].

6.2 State-Of-The-Art Evaluation Methods

In the early days of image retrieval, the Corel dataset was the first collection to be used for evaluation. It provided over 68,000 images, organized into classes of about 100 images, each with roughly the same topic. However, such artificial and relatively small datasets are not satisfactory as a benchmark nowadays [97]. We need to take into account different applications, the data they use (scope, size, metadata available, etc.) and the user information retrieval requirements.

The first serious effort for building a complex benchmarking platform for image search appeared in [118]. The proposed methodology was supposed to be realized by the Benchathlon[1] project, where research groups were meant to cooperate on creating the testing platform. Unfortunately, this project does not seem to be making any progress. Another analysis of image evaluation campaign can be found in [72]. It describes the background of ImageEVAL competition, which took place in 2006. However, the only repeated and successful benchmarking activity we know of is the ImageCLEF[2] [117] competition, which has been running since 2003. Each year, the organizers define various challenges, provide data and topics and evaluate the submitted results.

Nonetheless, even the ImageCLEF activities are limited by the availability of *benchmark inputs*, as defined in [110]: the data collection (documents),

[1] http://www.benchathlon.net/
[2] http://www.imageclef.org/

the queries (topics) and the ground truth (relevance judgements). We review these three issues in more detail in the following sections.

6.2.1 Image Databases

Gathering a large collection of image data is not a simple task due to the ownership and copyright issues. However, this can be overcome by using freely available web resources, such as the Flickr[3] web gallery or Wikipedia[4]. The following three datasets have been obtained this way. The first two have been composed to serve as benchmarking sets and are used in the ImageCLEF competition. All of them provide both images and text metadata, but they differ in size, origins and scope of available metadata.

- *MIRFLICKR collection*: The MIRFLICKR collection[5] [87] consists of 1 million images downloaded from the social photography site Flickr. All images are available under a Creative Commons Attribution Licence. The images have been selected based on their high *interestingness* rating that is determined by factors such as where the clickthroughs on the images are coming from, who comments on them, and whether they are marked as favorites. In addition, user-supplied Flickr tags, EXIF metadata and systematic image annotations are available. The visual descriptors provided are the MPEG-7 Edge Histogram and Homogeneous Texture descriptors, and the ISIS Group color descriptors.

- *Wikipedia collection*: The ImageCLEF 2010 Wikipedia collection[6] [133] extends the INEX MMWikipedia collection [162], which was created for the purpose of INEX evaluation campaign in 2007. Currently the collection consists of 237,434 Wikipedia images, their user-provided annotations, the Wikipedia articles that contain these images, and low-level visual features of these images. The collection was built to cover similar topics in English, German and French and it is based on the September 2009 Wikipedia dumps. For some images, no annotation is provided, other are annotated in one or several languages. Image visual features include both local (bags of visual words) and global features (texture, color and edges). The collection is available for the participants of the ImageCLEF competition.

[3] http://www.flickr.com/
[4] http://www.wikipedia.com/
[5] http://press.liacs.nl/mirflickr/
[6] http://www.imageclef.org/2010/wiki

- *CoPhIR image set*: The CoPhIR dataset[7] [16] with 106 million processed images is currently the largest collection available for scientific purposes. It consists of metadata extracted from the Flickr photo sharing system. The collection is composed mostly of outdoor and indoor photos, and there are also several images of e-shops products, cartoon images, hand drawings, paintings, etc. For each image, the collection contains a thumbnail image, a link to a corresponding entry at the Flickr web site, user-specified metadata (title, GPS location, tags, comments, etc.) and five MPEG-7 visual descriptors (Scalable Color, Color Structure, Color Layout, Edge Histogram and Homogeneous Texture). Since the data are not supervised, some of them are of poor quality - blurred or too dark/light images, images with sparse and erroneous annotations, different languages used in annotations, etc. While this may cause worse performance of search methods, the collection provides a good model of a real-world data.

6.2.2 Topics

The common goal of all search systems is fulfilling the information need of their users. Therefore, the test search topics should simulate what a real user would instantiate as a usage scenario. Furthermore, the volume (number) and diversity (variability) of the topics should cover the whole search domain and demonstrate statistical robustness of the results [110].

Usual ways of creating test search topics comprise a choice made by domain experts and an analysis of search system usage logs. In [133], the creation of topics for ImageCLEF 2010 Wikipedia Retrieval task is described in more detail. A candidate set of queries is derived from a search log file and topics from previous runs of the competition. From these, only such queries that have a sufficient number of relevant results in an organizers' search run are accepted.

Another issue related to the definition of topics is the choice of query modality. Usually, one of the following options is employed: query by example (image), query by text, or query by both text and images. Image queries are a natural choice for annotation tasks but are not suitable for image retrieval since one image may often represent several concepts, while the imaginary user is only interested in one of them. Therefore, either complex text queries or images complemented by text are used in web-like image search tasks.

[7] http://cophir.isti.cnr.it

6.2.3 Ground Truth

The term *ground truth* is used in information science to denote the best possible result that would be produced if a perfect method was applied in a given task. Results provided by different techniques are then compared to this ideal result. In classification tasks, a ground truth for a set of query objects will contain the correct class labels or, in case of multi-label classification, a binary decision about relevance of each label with respect to each query. In retrieval tasks, we may also desire to take into account the fact that some result objects may be partially relevant – not matching perfectly user's expectations, but also not quite irrelevant. This situation is better modeled by a ground truth that is expressed as a degree of relevance of a given object, e.g. as a percentage. In an ideal case, the ground truth should contain an indicator of relevance for each object in the dataset with respect to each search topic. To truly express the human-perceived quality of results, the relevance should be decided by human judges, preferably more than one for each object and topic to balance the subjectivity of opinions.

Clearly, creating such a ground truth is an enormously laborious process. When only a few people are involved, be them domain experts or lab members, providing exhaustive relevance judgements is only feasible for relatively small datasets. For the large ones, some approximations are usually employed. The following sections detail different techniques that are being used to collect the ground truth data.

Because of the difficulties related to ground truth acquisition, evaluation campaigns such as ImageCLEF do not provide the evaluation data to the research community, but keep them to be used in future competition runs. In consequence, there is a deficiency of ground truth data for testing outside the evaluation campaigns, which is definitely an obstacle in the development of new search methods.

Expert Evaluations

The most straightforward way of collecting the ground truth data is asking a group of expert users to evaluate the relevance of all object-query pairs. However, this is only feasible for small datasets. Alternatively, expert annotations can be provided for each image, using a defined categorization. This is also a tedious work, but only needs to be done once for each object in the dataset. However, the keyword ground truth can only be used for evaluation of classification tasks. This approach has been adapted by the supervisors of the MIRFLICKR dataset [87].

Automatic Ground Truth Extraction

As opposed to the expert evaluations, automatic ground truth extraction techniques seek to provide the relevance data without user involvement. Typically, this approach is also used for classification task evaluation. Starting with a set of categories, various algorithms try to identify the relevant images in selected data sources (web, web gallery, etc.) and download them to form the test dataset. Naturally, the images are accompanied by the category label. Various information sources such as ontologies and text corpora can be combined to improve the precision of automatic identification of images belonging to a given category [142, 155].

Pooling

As it is very difficult to collect the complete ground truth data with large datasets, a *partial ground truth* is often utilized in large-scale evaluation. The partial ground truth only contains relevance assessments of object-query pairs that are needed for a given evaluation task. In most evaluation campaigns, the objects for the partial ground truth are selected by so-called *pooling*: objects that appear among the top n images in any of the results submitted by the competitors form a so-called *pool*, and expert evaluations are provided only for the objects in this pool [133]. This approach allows fair comparison of a given set of results, but only provides a one-time ground truth that cannot be meaningfully reused for evaluation of different result sets.

Crowdsourcing

The only way to obtain an exhaustive ground truth for large datasets is by employing a large group of people, which is often denoted as *crowd* in modern information retrieval. However, it is not easy to find the necessary motivation for the crowd to preform the requested task. One possible approach is to invest a considerable amount of money and pay for each judgement. This approach was adapted to create the ImageNet database [56], where the Amazon Mechanical Turk platform was used to manually clean a large set of candidate images. Similarly, crowdsourcing was utilized to obtain ground truth data for some of the ImageCLEF tasks [126].

Naturally, money is not the only available means of motivating users to cooperate. People can provide information about images in return for some service or simply for fun. The authors of the TagCaptcha image annotation system [114] propose to obtain annotations via the widely used Captcha

challenge-response tests. Users are asked to choose relevant tags for several images, some of which are known to the system but other are not. When the known tags are assigned correctly, the other tags are learned for the untagged images. In a similar manner, the famous ESP game [157] was designed to entertain users and collect image labels at the same time.

Crowdsourcing is a powerful tool that allows to obtain a lot of data rather easily. However, it needs to be handled carefully. Recent book [84] explains the need to choose the crowd wisely, so that the people are well-motivated and provide relevant data. Naturally, it is also necessary to check the collected data for inconsistencies. Otherwise, the data obtained by crowdsourcing may be useless. In [31], we detail some of the problems we encountered in crowdsourced data provided within the data of ImageCLEF 2011 Image Annotation Task.

6.3 Profiset Evaluation Platform

To be able to assess the effectiveness and efficiency of the search methods that focus on large-scale retrieval in broad domains, we needed a large test collection that would be equipped with a rich ground truth, preferably in the form of degrees of relevance. Due to the lack of freely available evaluation test-beds that would meet these requirements, we decided to create our own data collection and associated ground truth. To achieve this, it was necessary to collect a considerable amount of relevance assessments, provided by human judges. To make this task feasible, we proposed a novel method of ground truth acquisition which is based on the crowdsourcing approach but relies on cooperation of research groups rather than financial motivation of participators.

6.3.1 Dataset

Our objective is to provide a dataset that will enable researchers to test systems for large-scale searching in terms of results quality (precision), efficiency (search time) and scalability. The important aspects are therefore (1) the size of the dataset, (2) its scope, and (3) the type of data provided. As to the size, the datasets that are used for benchmarking nowadays range in volume from a hundred thousand to several millions of images. We believe that even larger datasets are necessary to test the efficiency of methods for web searching. Regarding the scope, we are interested in a real-world dataset since the performance of search mechanisms is influenced by the distribution of objects in the domain. Finally, recent research indicates that

the future of image searching seems to be in combining multiple modalities, typically visual features and text metadata describing the semantics. Therefore our dataset should contain at least these two modalities.

The Profiset collection which we chose as the basis of our evaluation platform satisfies all the discussed requirements. The images were kindly provided by the Profimedia[8] company, who sells stock images produced by photographers from all over the world. The collection contains 20 million high-quality images with rich and systematic annotations, thus providing rich data in both visual and text modalities. For each image, we have extracted five MPEG7 [116] global visual descriptors recommended in [16]. The dataset is freely available for research purposes and provides the following information for each entry:

- a thumbnail image;
- a link to the corresponding page on the Profimedia web-site;
- two types of image annotation: a title (typically 3 to 10 words) and keywords (about 20 keywords per image in average) mostly in English (about 95 %);
- five MPEG-7 visual descriptors extracted from the original image content: Scalable Color, Color Structure, Color Layout, Edge Histogram and Region Shape.

6.3.2 Query Topics

When selecting the topics, we had the following requirements in mind: the queries should reflect real users' needs, the topics should be diverse both in content and in complexity, and there should be enough relevant results for each test query in the dataset.

To achieve this we first created a set of candidate topics which comprised (1) popular queries from the search logs provided by Profimedia, and (2) several examples of queries that we know from experience to be either easy or difficult to process in content-based searching. Next, we run a top-30 query for each of the candidates, using different combinations of text-based and content-based search methods discussed in Chapters 4 and 5. Only the topics for which at least 10 relevant results were found were accepted into the final query set.

[8] http://www.profimedia.com/

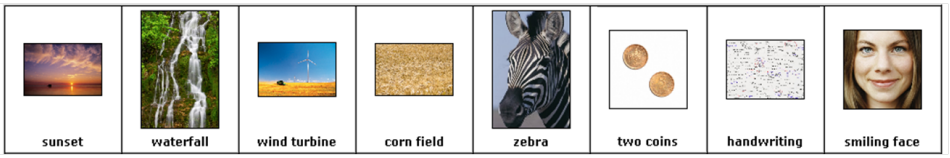

Figure 6.1: Query object examples.

The resulting set of test queries consists of 100 topics, each of which is defined by a single query image and a few keywords (typically one or two). The following categories are represented by the topics: activity (5 queries), animal (8), art (6), body part (5), building (3), event (3), food (8), man-made objects (16), nature (16), people (12), place (9), plant (2), specific building (4), and vehicle (3). Several examples of the query objects are shown in Figure 6.1.

6.3.3 Partial Ground Truth

As stated earlier, a full ground truth should contain relevance evaluation for each topic-object pair. However, collecting a full ground truth for a large dataset is only feasible when a lot of people are employed. Since we lack the resources required to organize such a campaign, we have utilized a combination of the pooling approach and crowdsourcing. In particular, we created a pool of candidate images and asked our lab colleagues to act as the judges, thus creating a partial ground truth for the set of test queries we selected. The initial pool was prepared in such a way that it should cover the majority of relevant images. Furthermore, we provide tools that allow other researchers to expand the ground truth as needed.

Pooling Data

In benchmarking competitions, the pool of candidate images for the evaluation usually contains the top n objects from each submitted result. We applied a similar technique but we have used a set of our own search methods implemented over the MESSIF framework [21] that provided the results. In particular, we have employed all methods discussed in Section 5.3 that represent various existing approaches to retrieval by text and visual modalities, and several postprocessing techniques that exploit the pseudo-RF paradigm. Human-assisted query expansion was also applied to obtain even more candidate objects. The various solutions that we have used are schematically depicted in Figure 6.2.

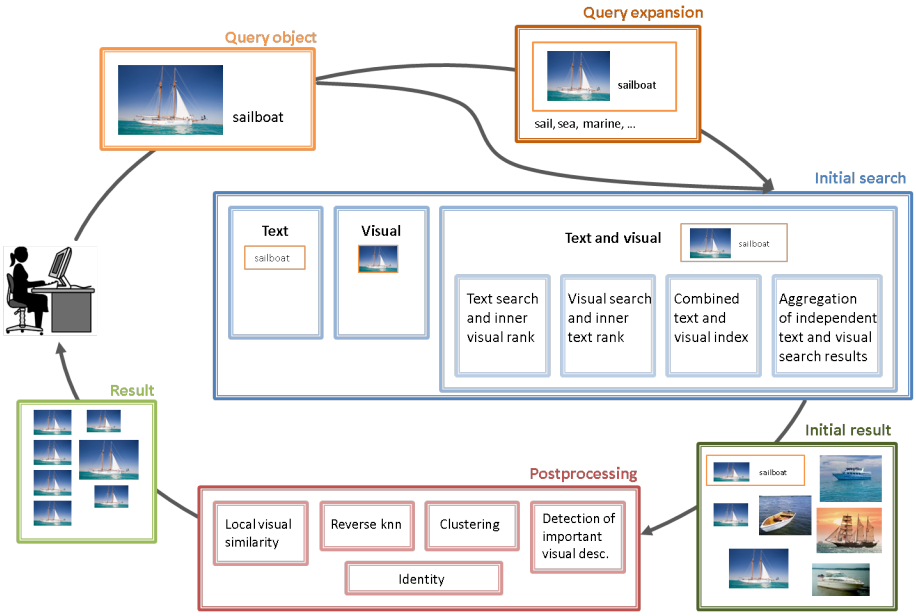

Figure 6.2: The global search schema.

Relevance Judgements

Altogether with variable weights settings, we created 140 search methods. For each query image, top-20 queries were evaluated by all these methods. The results were then merged and displayed in a web interface, shown in Figure 6.3. The judges were asked to mark each object as *very good*, *acceptable*, or *irrelevant*. To obtain the overall relevance score of a result object with respect to a given query, the relevance categories can be transformed into numerical values and aggregated as needed.

Statistical Evaluation

The ground truth data we obtained from our judges contain a considerable number of relevance evaluations which are a valuable resource for analysis of human perception of similarity. In this section, we present several observations concerning both the properties of our dataset and the human factor in the evaluation. The analysis is based on the ground truth snapshot from summer 2011, when we prepared the publication [30] about the Profiset platform. Since then, the number of evaluated objects has significantly increased during the evaluation of many additional experiments.

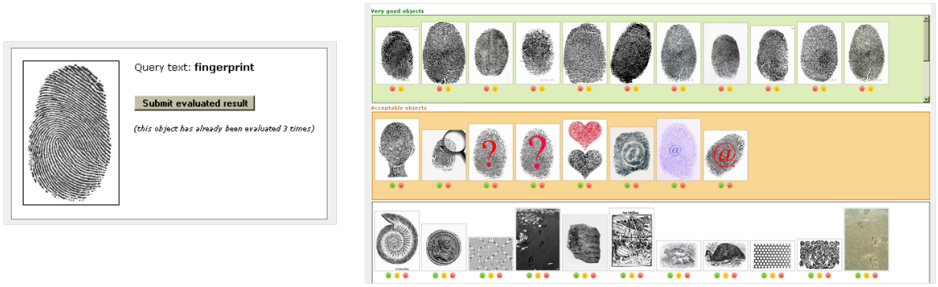

Figure 6.3: The web interface for relevance evaluation.

The original evaluation process was performed by 15 participants, most of them students, graduates, or researchers in IT. Out of the 100 queries, each got evaluated at least twice, the total number of evaluations being 222. With the average number of candidate objects per query topic being 578, we obtained a total of 128,141 evaluated topic-object-user triplets. The evaluation process took a month, the actual time invested in the judgements being about 100 hours.

As mentioned earlier, we can compute the relevance of a result object as the average of all evaluations we have for it. We find it suitable to categorize objects into the following categories: *perfect* (average relevance 100 %), *good* (at least 50 %), *partially relevant* (more than 0 %), and *irrelevant*. For each query topic in our testbed, there were in average 105 perfect result objects, 223 good objects and 315 irrelevant ones. However, the number of objects in each category differed considerably between individual queries – the lowest number of perfect results was 5, and 11 objects had less than 20 perfect results. The lowest number of good results per query was 53. We can conclude that our set of topics is suitable for testing as there are enough relevant objects to be found and, at the same time, enough queries with various difficulty levels are present (difficulty being inversely proportional to the number of relevant objects contained in the dataset).

When evaluating the results, the judges were not given any rigorous instructions on what shall be considered relevant. Therefore, their classification of results reflects their individual understanding of similarity and their expectations of image search system performance. While this is known to be subjective and inconsistent in different situations, all image retrieval systems are based on a tacit assumption that there exists some basic agreement in the individual opinions. Using our relevance evaluations, we can verify this assumption. Table 6.1 shows the percentage of identical evaluations, where all judges agreed on the (ir)relevance of a query object pair.

Number of evaluations	Identical evaluations	Unmatched (2 different)	Unmatched (3+ different)
2	80 %	20 %	–
3	70 %	27 %	3 %
4	73 %	21 %	6 %
5	65 %	20 %	15 %

Table 6.1: User agreement in relevance assessments.

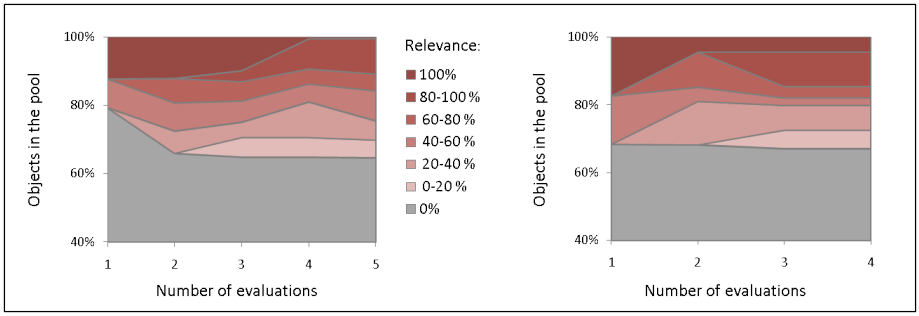

Figure 6.4: The development of result evaluations for two different queries.

For the sake of our ground truth it is also important to know whether two judgements (which we have for most queries) are sufficient to obtain a trustworthy relevance evaluation or whether more opinions are needed. Figure 6.4 shows how the percentage of objects with given relevance changes with the growing number of evaluations (we used the results with the most evaluations to obtain these graphs). We can observe that the results are quite stable, therefore the two judgements can be considered sufficient.

Finally, let us have a look at the methods we used to create the candidate pool. We employed a high number of combinations of search methods and postprocessing techniques in order to discover as many relevant objects as possible. This approach has proved to be well suited as every combination did bring some relevant object to the results that was not found by any other method.

6.3.4 Provided Functionality

As we already mentioned, there is a lack of available testbeds for evaluation of large-scale image retrieval. To help the research community to overcome

this problem, we designed the Profiset collection from the beginning to be publicly available, easy to use, and extensible. The dataset can be downloaded from `http://mufin.fi.muni.cz/profiset/` after registration and agreement to the usage terms. The data can be freely used for research purposes.

In order to offer the tools created during the preparation of the partial ground truth for the research community's benefit, we have designed two web-services. The first one simplifies the benchmarking of an external search method against the existing ground truth and provides means for extending the partial ground truth. The second service allows to add a new image to the query set and collaboratively evaluate its ground truth.

Evaluation of External Search Method

Researchers proposing new search methods for image retrieval systems are welcome to use the Profiset testbed as a benchmark. By downloading the dataset, the query set, and the ground truth, they can compute their own statistics on the effectiveness of their method. However, since our ground truth is only partial, the new method may retrieve images whose relevance is unknown.

To overcome this problem, the results of the new method can be uploaded to our service. In particular, only the identifiers of the images found relevant by the new method for a particular query (from the set of test topics) are uploaded, so there is no need to adjust the implementation of the method in any way. The service then checks all the objects that were in the original candidate set from which the ground truth was computed and any new image is presented via the web-interface. Users are then able to judge whether each of the new objects is *very good*, *acceptable*, or *irrelevant* in the same way as when the previous partial ground truth was created. Afterwards, the statistics of the new method using the updated partial ground truth are displayed.

Any such addition to the existing partial ground truth is also stored in our database and immediately available for download. Thus, the partial ground truth is collaboratively extended whenever a new method is tested via our service.

Additional Query Images

Since our query set consists of a hundred images while the dataset contains 20 million images, we offer a service that allows to introduce additional

query topics. Naturally, it is also desirable to provide some ground truth information for any new query. In order to do that, we need a candidate set of images and then user judgements of the relevance of the respective images (as explained in Section 6.3.3).

Our service thus allows to upload a new query topic (possibly by selecting an existing image from the Profiset collection using its identifier). Then one or more candidate sets can be uploaded, e.g. retrieved by some new search methods (as in the other service above). Finally, the system asks whether the candidate set should be expanded by our search methods. We provide options for selecting our text-based, content-based, or combined methods as explained in Section 6.3.3. Since the candidate set creation is a computationally intensive task, a job is scheduled in our university GRID[9]. It can take some time until the candidate set is ready for the user judgement, so the service notifies the user by email.

Then, the new query object is available for the user evaluation via a web interface as shown in Figure 6.3. When at least one evaluation is complete, the query is available in the query set with the new partial ground truth. The query is then also offered for additional evaluations to other users.

Fair Use

We expect that researchers will use the service in good faith. However, since the access to the system is authenticated, we are able to identify possible abuse of the system and remove incorrect judgements. We also distinguish the candidate objects based on the method that selected them, so it is easy to filter out the results from an "invalid" method.

6.4 Summary

As we have demonstrated in the first part of this chapter, there is a lasting need for open evaluation platforms for multimedia retrieval. In particular, no large test-beds are available for image retrieval testing apart from a few evaluation campaigns, which are not suitable for continuous development of search methods. Therefore, we have proposed and created a new evaluation platform, which contains 20 million real-world images, a set of test queries, and a partial ground truth. We also provide supportive web services that allow other researchers to access the test-bed easily and cooperate on extending the ground truth. We believe that the research community

[9] http://www.metacentrum.cz/

can act as a specialized crowd motivated to provide high-quality relevance data.

The Profiset evaluation platform as well as the methodology of its creation were presented in [30]. As of December 2012, 8 users of the platform are registered and the ground truth contains more than 140.000 evaluated query-result pairs. Using the test-bed, we were able to perform the exhaustive evaluation of multi-modal search methods presented in Chapter 5. Moreover, the proposed methodology can be readily applied to create additional test-beds, utilizing a different dataset or targeting a different application. A sister CoPhIR evaluation platform already exists that utilizes the CoPhIR dataset, and the Profiset Annotation test-bed provides means of testing automatic image annotation. Both of these were used in experimental evaluations reported in this work (in Chapters 5 and 7) but are not publicly available at the moment.

Chapter 7

Applications

The query-by-example image retrieval which we have studied in previous chapters can be utilized as a stand-alone tool in many tasks, such as targeted search, collection browsing, or detection of image (near-)duplicates. However, there is also a number of scenarios in which retrieving similar images forms only one part of a more complex processing that is needed to obtain a desired piece of information. In such situations, visual similarity between images typically provides a link to relevant information of some other type, which is then processed by the application.

One of the most frequent procedures that utilize content-based searching is automatic image classification or annotation. Automatically generated textual data that describe image content find use in numerous contexts, ranging from librarian classification of multimedia items to free text descriptions of a personal photo collection. Even though the classification tasks are traditionally associated with machine-learning techniques, search-based annotation mining is also beginning to gain attention, especially in applications dealing with broad domains and unlimited number of categories (labels). As the requirements on image retrieval may differ in the context of text mining and in general image search applications, we need to study the behavior of retrieval techniques also in the context of image annotation applications.

Accordingly, the image retrieval methods developed within the MUFIN search system were applied and tested in two scenarios that strive to mine textual information from images. The MUFIN Annotation Tool application aims at providing a set of descriptive keywords for an arbitrary input image, exploiting the descriptions of similar images. This tool was first published in 2011 and further refined in 2012, when it was also integrated into a Firefox plugin, thus becoming more comfortable to use and freely available to web users. In 2011, we also participated in the ImageCLEF contest and developed a search-based solution for its image categorization task. To decide the relevance of categories, we exploited the free-text annotation provided by the MUFIN Annotation Tool.

In this chapter, we first provide a brief introduction of relevant use cases and a survey of existing approaches to image annotation. Special attention is given to the principles of search-based annotation and identification of the main challenges in this area. The following sections describe the above-mentioned applications of the MUFIN search engine to annotation and classification tasks. Building upon the recent work in the annotation research, we create a general model of the search-based annotation, provide a working implementation, and lay foundations to a systematic study of search-based text mining. Current results as well as future research plans are summarized in the final section.

7.1 Extracting Words From Images

Mining text data from images has recently become a very popular research field, comprising a wide variety of tasks with different challenges. Before introducing the main directions of contemporary research, we would like to present a few use cases that illustrate the various information needs that are being studied.

- *Cell type recognition* is a textbook example of a simple classification task. A cell image needs to be classified into one of several classes. Correctly labeled training data are usually available.
- *Multimedia archiving* applications typically need to sort data objects into classes, but this time more than one label may be relevant for a given object. However, the set of labels is limited and known.
- *Text-based image retrieval* is comfortable for users but requires images to be richly annotated. To support text-based retrieval, image annotation methods should provide as many relevant tags as possible for a given image. The application domain may be very broad and the set of labels potentially unlimited.
- *Semantic web* philosophy requires image annotation methods to supply not only relevant tags but also links to relevant sources of semantic information, e.g. ontologies. Semantically annotated multimedia can then be exploited in complex information mining applications.

7.1.1 Overview of Approaches

To select a suitable approach for any of the scenarios suggested, it is important to consider the following factors: the level of user involvement in the

annotation process, input data properties, the target vocabulary, and the performance issues.

Naturally, the annotation is more precise but much more labor-intensive with user involvement. Research of semi-automatic annotation methods focuses on simplifying this task by providing advanced annotation interfaces [101, 153]. Also, the crowdsourcing paradigm can be utilized, exploiting game-like methods as discussed in Section 6.2.3. A comparison of different approaches to manual image annotation can be found in [107].

On the other hand, automatic approaches can only work with the information present in the input data. More specifically, an annotation algorithm needs to extract the relevant data from the query image and some knowledge base, which is typically a collection of labeled images. In some cases, additional metadata such as GPS coordinates may be provided with the input image, which can be used with advantage e.g. by landmark recognition applications. Alternatively, the surrounding web page can be analyzed to learn about the image content. In many situations, however, the image itself is the only input of the annotation process. Then, it can be processed in the following ways:

- Model-based image annotation: A correctly labeled training dataset is used as an input for machine learning processes, which create a statistical model for each concept from the annotation vocabulary. Local image descriptors are typically clustered into visual words (blobs) and the machine learning techniques establish relationships between the visual words and text labels. These are then used to provide the annotation.

- Search-based image annotation: The annotation process starts with a content-based search in the knowledge base, which retrieves a set of images similar to the query image. The texts associated with the similar images are consequently analyzed and the output annotation is composed, e.g. by selecting the most frequent keywords.

The suitability of the model-based or search-based approach is tightly related to the type and scope of the target vocabulary of the annotation. The survey study [78] distinguishes three types of annotation: free text, keywords chosen from a controlled dictionary, and concepts from an ontology. Annotations of the first type are not restricted in any way, while the other two types presume some knowledge about the domain and task. Ontologies are particularly important for semantic annotations and will be discussed in more detail later. Another important aspect is the size of the

vocabulary, which ranges from a couple of class tags to the size of natural languages. In case of smaller vocabularies we often speak of *classification* rather than *annotation*. By their design, model-based methods are better suited for classification tasks with a limited and controlled vocabulary, where a separate classifier can be learned for each concept. On the other hand, search-based approaches are more appropriate for free-text annotations in broad domains, such as web image tagging. Naturally, the basic models can be adjusted and combined in numerous ways.

The last important topic we need to consider in the context of image annotation is the efficiency of available methods. Most of earlier work on image annotations focused on small-sized static datasets, where the efficiency issues are not relevant and the main challenge is to tune the classifiers. However, the attention of the research community has recently shifted to comprehend also the large-scale annotation problems, which often require online processing of large quantities of data. Naturally, this again results in the need to carefully balance the precision and costs of both the image retrieval phase and the text mining procedures. Even though promising results have already been presented for large-scale annotation [99, 150], a lot of challenges still remain open. We discuss them in the following sections, which survey the latest development of annotation methods in three directions: development of ontologies and their usage for annotation, utilization of model-based approaches in large-scale annotation, and advances in search-based approaches.

7.1.2 Ontology-Based Annotation

Ontologies as a tool for describing objects (both real-world and abstract) and their relationships are a promising source of semantic information that can be exploited in automated processing of multimedia data. Even though state-of-the-art ontologies are only able to provide the semantic information for a few selected domains, we believe knowledge bases of this type will be an important part of information mining in the future. Therefore, we briefly examine existing ontologies for image description and their utilization for annotation.

In the context of visual information, ontologies can be designed to describe various aspects of an image: thematic descriptions of depicted matter (scene, objects, events, etc.), media descriptions referring to low-level features (descriptors, extraction algorithms), or structural descriptors (segmentation of image, spatio-temporal aspects) [53]. Currently, a lot of research groups and initiatives focus on designing multimedia ontologies

with respect to the latter two issues [153], whereas the thematic ontologies for image content are more difficult to find. The majority of existing thematic ontologies focuses on specific application domains, e.g. biomedical data [52]. However, general ontologies for multimedia annotations are also beginning to appear, such as the basic "Photo Tagging Ontology" used within ImageCLEF annotation tasks in last years or the more complex Large-Scale Concept Ontology for Multimedia (LSCOM) [119] which has been cooperatively developed for video news annotation. Still, the WordNet [69] lexical database remains the resource most frequently used for retrieving semantic relationships of depicted objects. Even though the WordNet, strictly speaking, is not an ontology, it is the largest available collection of semantic concepts interlinked by relationships. The WordNet is exploited e.g. in [121] to extract semantic meanings of images from the unstructured textual annotation provided by web authors, or in [92] to prune irrelevant keywords from the classification results. In a similar way, Wikipedia pages and their categorization have been used as another source of semantical information.

7.1.3 Machine Learning

Machine learning techniques represent a standard model-based approach to image annotation and classification. The survey study [166] introduces main research directions in this area, which include support vector machines, artificial neural networks, decision trees, and Bayesian methods. A wide variety of methods have been proposed for each of these approaches and to analyze them is beyond the scope of this work. However, an important property common to all machine-learning techniques is the need for correctly labeled training data. Also, the model-based approaches are only able to distinguish a limited number of image classes, for which the classifiers are available. To apply this paradigm also for annotations with large vocabularies, it is necessary to use a lot of classifiers and expand the classification results.

In the context of large-scale annotation, a well-known example of model-based solution is the ALIPR [99] system, which claims to provide real-time annotations for web images. ALIPR uses the Corel dataset with about 600 semantic concepts as a training dataset, each concept being described by several words. After the classification, words from the most relevant concepts are merged to form the annotation. A similar approach is proposed also in [54]. Different solutions to the ImageCLEF annotation tasks [126, 128] elaborate on the selection and combinations of visual descriptors for

classificators but do not perform any subsequent processing of selected concepts.

7.1.4 Search-Based Annotation

Search-based image annotation, also denoted as data-driven annotation, is an orthogonal approach to machine learning [166]. For this solution, no previous training of the annotator is needed. In time of the query execution, a content-based image search is initiated over the knowledge base, which retrieves a set of similar images. The text metadata of these images are subsequently exploited to provide the annotation. In this basic setting, vocabulary of the annotation reflects the vocabulary of the reference dataset. If required, the vocabulary can be adjusted with the use of dictionaries, ontologies, etc.

From a number of recent search-based solutions, we select a few that illustrate different adjustments of the basic processing schema. In [159], the authors elaborate on a ranking algorithm that selects the annotation keywords from among the candidates using a random walk in a graph of tag associations. Authors of [150] propose an innovative way of obtaining the reference dataset. Issuing a text query to several web search engines for each non-abstract noun from the WordNet dictionary, they collect 80 million web images, each of them being associated with one noun. The images are further downsampled to 32 x 32 pixels to save processing costs and successfully used for recognition of objects and scenes. In the final phase of the annotation process, WordNet is employed again to remove labeling noise. The AnnoSearch system [161] follows a different strategy, requiring the annotation process to start with the query image and a seed keyword. Using the keyword, text-based queries are employed to provide candidate images. These are subsequently ranked by visual similarity and the top ones provide keywords for the annotation. In [60], the semantic annotation corpus was prepared offline by mining 400K Web images and their surrounding web pages. The images were found by text search using concepts from the LSCOM ontology. The resulting corpus provides *class names* and *class keywords*, which are employed in the following way: content-based retrieval selects 200 most similar images, 5 most frequent classes are identified among these, then their keywords are added to the candidate set and the final relevance of each annotation word is evaluated. A very recent study [42] suggests that different approach should be used for *scene* and *object* tags. The authors argue that scene tags can be retrieved by standard search-based approach, but object tags are more difficult to obtain. To solve

the problem, the authors propose to build a concept dictionary for the object tags.

Challenges

Search-based image annotation is a promising direction for further research as it allows to exploit the vast amounts of user-generated data available on the internet. Taking advantage of recent progress in content-based multimedia searching, it offers a complementary approach to traditional machine-learning techniques. The following list presents current open challenges for search-based annotations:

- Choice of data sources: The reference dataset used for content-based retrieval significantly influences the quality of annotations. One of the important factors is the size of the dataset and its scope – the broader the domain to be covered is, the more images are needed to provide examples for different visual forms of the semantic concepts. Current solutions comprise utilization of existing collections, typically provided by web galleries such as Flickr, or creating a dedicated image collection from web images selected by text search, using keywords from different sources (WordNet, ontologies) as seeds. In the latter case, another design decision concerns the selection of tags for the knowledge base – either only the seed keywords can be utilized, or a whole web page surrounding a given image is analyzed to provide richer vocabulary. Denoising of the knowledge base can also be performed, using ontologies and word co-occurrences.

- Text data processing: The labels associated with similar images form the basis for annotation formulation. In this process, relevant keywords should be identified and if required, transformed to match a given vocabulary. Often, it is also advisable to try to expand the annotations by related words from other sources to eliminate the problem of a closed vocabulary in case of static reference databases. A wide range of techniques have been proposed for this task, mostly exploiting statistical properties of the reference dataset (keyword frequencies, associations between tags) or external resources (WordNet, ontologies).

- Efficient and effective annotation: To provide a real-time annotation, the performance of both the image retrieval and the metadata processing is vital. This is affected by many factors, including the visual

content descriptors in use, the knowledge base volume, the data indexing and search algorithms employed, and the complexity of text metadata processing. Recent research works present different solutions for each of these issues but to the best of our knowledge, no systematic study of annotation efficiency has been carried out yet.

- Evaluation: Similar to other information retrieval tasks, evaluation is a crucial part of the development of annotation techniques. As well as in the case of image retrieval, the difficulties arise with evaluation of large-scale tasks. Some evaluation datasets already exist, e.g. the IAPR TC-12 Benchmark collection [76] or the NUS-WIDE dataset [44], various other methods are discussed in [78]. A common problem of many evaluation datasets is the limited vocabulary of available ground truth, which constrains the usability of such testbeds for evaluation of free-text annotation methods.

7.2 MUFIN Image Annotation

Search-based image annotation is a promising and active research direction that is a natural extension of our previous research in the field of image retrieval. Taking advantage of having a working image search system, we can now concentrate on the specific issues related to text information extraction. In particular, we focus on the situation when an unknown image file is the only input and a set of relevant words from an unlimited vocabulary is expected on the output. We assume that a real-time interaction with the annotation system is needed. Such application would find use e.g. for tag hinting in a web gallery or a photostock site.

This section discusses the design and development of the MUFIN Annotation Tool, which was created to address such use cases. The functionality of the MUFIN Annotation Tool is composed of two distinct phases – the retrieval of similar images, and the processing of text metadata of these images. We introduce our current solutions for both these phases, evaluate their efficiency, and analyze the results.

7.2.1 Annotation Framework

Figure 7.1 presents a general model of a search-based annotation system. Its principal components can be found in all systems surveyed in Section 7.1.4. The individual subtasks represent important issues that need to be thoroughly studied before a real working annotation system can be created. As

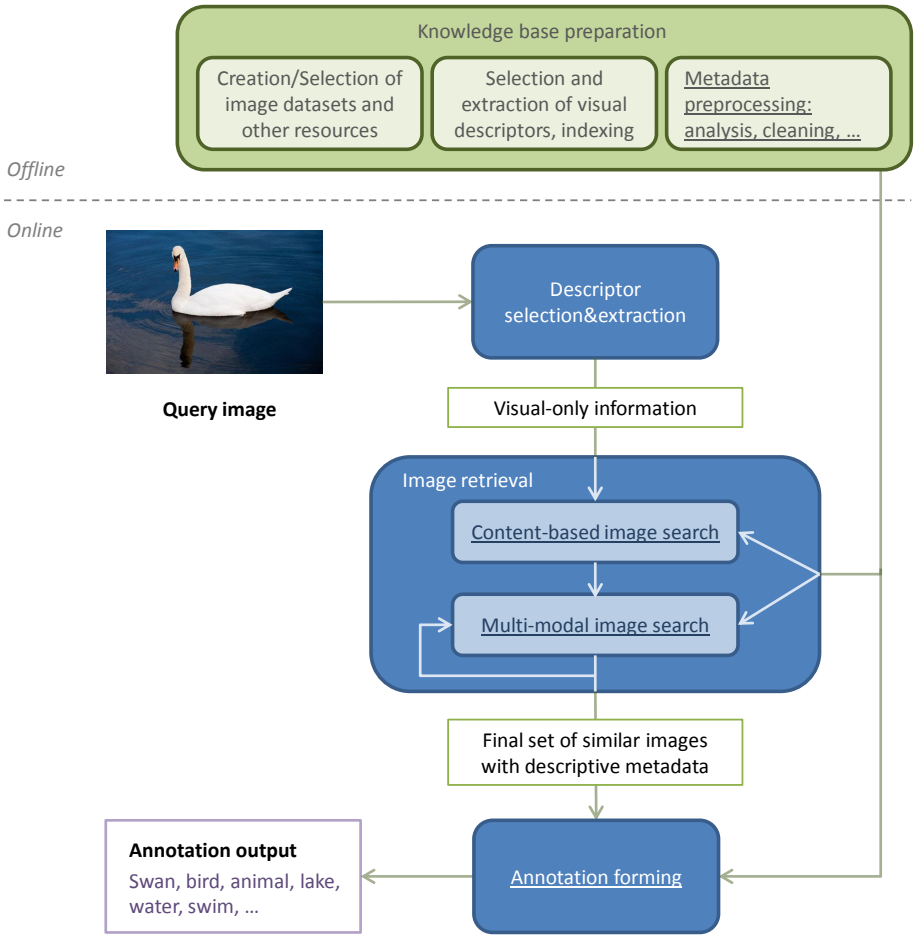

Figure 7.1: Data-driven annotation model.

we shall see, some basic annotation functionality can be achieved relatively easily. However, the results are far from satisfactory for real applications. Therefore, a lot of research effort still needs to be invested to understand the potential of different data sources and mining techniques and utilize them optimally. In our current work, we focus on optimization of the components which are highlighted in Figure 7.1 by underlining.

7.2.2 MUFIN Annotation Tool

The MUFIN Annotation Tool first came into existence as a simple application for testing the performance of our image search methods in the annota-

tion use case. After preliminary testing of the first prototype, which showed promising results, we continued refining the image searching as well as the text mining techniques.

To build even the most basic tool capable of transforming images into words, we needed to face the following issues: 1) find an image dataset with text metadata that is large and precise enough to serve as a knowledge base; 2) choose some measure of visual similarity that is applied to select images similar to the query; 3) facilitate content-based retrieval; and 4) decide how the final annotation should be derived from the metadata of the similar objects. For the first version of the Annotation Tool, we solved these issues in the following way:

- We took the advantage of having a large, reliable source of annotated images in the Profiset collection (described in Section 6.3.1) and chose it as a knowledge base.
- The standard combination of five MPEG-7 visual descriptors used in MUFIN [16] was employed to evaluate the visual similarity of objects.
- Content-based search such as described in Section 5.1.3 was utilized to find images similar to the query image.
- The final annotation was composed of the most frequent keywords from the descriptions of the similar images.

We used this first prototype to gain some experience with the behavior of annotations and to identify the directions for future development. Also, this implementation was utilized in the ImageCLEF Annotation Task, which will be discussed in detail in the following section.

As the next step, we tried to improve the informative value of the set of similar images by combining the results of content-based search over several diverse data collections. This approach has a strong rationale, as it prevents the annotation vocabulary from degradation and allows to exploit information from sources with different origins and biases. Therefore, we expanded our implementation by adding another knowledge source, namely the Imagenet collection [56] which tries to collect images illustrating the WordNet concepts. At the time we used it, the ImageNet dataset contained about 12 million images. The prototype of the Annotation Tool in this phase, as shown in Figure 7.2, was presented in a demo paper [32] at the SISAP 2011 conference. However, the ImageNet dataset did not prove to work well, as its text metadata are too sparse. Since we were not aware

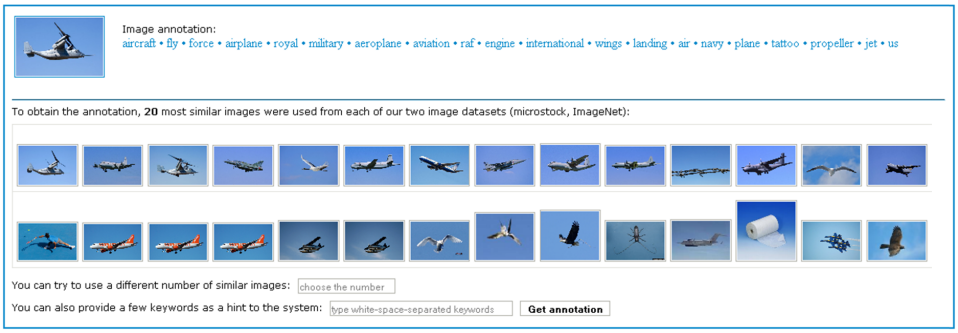

Figure 7.2: MUFIN Annotation Tool (as of 2011).

of any other promising image dataset, we abandoned this research direction temporarily, even though we do believe that combination of multiple image sources is a promising direction for the future.

After the initial experiments, we decided to focus more thoroughly on the following challenges:

- Utilization of multi-modal retrieval for selection of similar images: At the beginning of the image retrieval phase, only visual descriptors of the query image can be exploited for searching. Still, different visual features can be used and combined in various ways. After the initial search, derived modalities also become available. We study a few selected multi-modal image search scenarios and their behaviour with different parameter settings.

- Preparation of the knowledge source: Even though we employ a high-quality image dataset, there is still a significant level of noise in the image descriptions. Therefore, metadata cleaning tools are needed. Moreover, we try to establish links between the text metadata of Profiset images and the WordNet synsets, which would allow us to exploit the semantic links in the WordNet hierarchy during the annotation process.

- Selection of annotation keywords: By exploiting the links between Profiset keywords and the WordNet hierarchy we try to improve the selection of relevant keywords.

In the following sections, we discuss our current solutions for each of these issues. Subsequently, the usefulness of the proposed techniques will be analyzed in an experimental evaluation.

Retrieval of Candidate Images

Even though various image descriptors have been tried in data-driven image annotation, there is no clear consensus on which are the most suitable. In the current phase of our research, we therefore utilize the standard descriptors approved in many image-search-related applications – the MPEG-7 global descriptors and the SIFT local descriptors. As we have discussed in Chapter 3, the retrieval by local image descriptors is, in general, more costly and therefore is less eligible for fast searching in large image collections. The annotation response time is however one of the important qualities in interactive applications such as the tag hinting. Therefore, we exploit the global visual features in the initial retrieval of similar images, in particular our usual fixed combination of five MPEG-7 descriptors (more details are provided in Section 5.1.3).

After the initial search, which provides the first links to the knowledge base, it is possible to apply an unlimited number of query expansion steps and additional search runs to provide the best possible set of similar images. Naturally, the additional steps increase the processing costs as well as the complexity of the whole system. To be able to combine multiple search runs effectively, it is necessary to have a deep understanding of the possibilities and limits of each processing phase. In the current version of the MUFIN Annotation Tool, we restrain our solution to a single search with subsequent postprocessing techniques and focus on its optimization, thus establishing the foundations for future enhancements.

As we could observe in experiments presented in Section 5.3, the results of content-based retrieval by global descriptors can be significantly improved by subsequent re-ranking of the basic search results by a complementary modality. In this case, however, the secondary modality is not explicitly contained in the query specification and needs to be extracted from the query image. Alternatively, pseudo-RF re-ranking can be employed, which exploits information from the candidate results. Related literature as well as our preliminary experiments suggested that there are two promising approaches that could be employed: 1) re-ranking by local visual similarity, using SIFT features extracted from the query object and the candidates, and 2) pseudo-RF re-ranking that exploits the clustering principle, i.e. ranks the candidate objects by their average distance to other objects in the candidate set. These two methods were therefore studied in the context of image annotation. Ranking by the SIFT descriptors allows us to include an orthogonal view on image similarity with relatively low additional costs. With the clustering rank, it is of a particular importance that also the text

descriptions of candidate result objects are taken into account, as both text and visual modalities are available for these objects. Thus, the final result is composed of objects mutually similar on both visual and semantic levels, even though a visual-only query was issued.

A crucial factor for the precision of the image retrieval phase is also the choice of the result sizes after each similarity search or postprocessing step. We strive to optimize these parameters to obtain the best possible ratio of relevant and irrelevant results after each step, which increases the probability of producing high-quality results in the next step.

Text Preprocessing

In some aspect, the utilization of web resources for the data-driven annotation is very promising as it allows us to access vast amounts of information contributed by internet users. It would be hardly feasible to collect such quantities of data in a more controlled environment. On the other hand, the web resources also inherently contain problematic issues that arise from the fact that the data was originally intended for a different use than the annotation. Tag noise and tag ambiguity are two of the most obvious issues related to text metadata of image objects. To eliminate these problems, data cleaning and disambiguation should be applied before a data collection is employed in the annotation process.

While the problem of text metadata cleaning is closely related to the well-studied issue of text data preprocessing for text retrieval, it has some specific features which prevent us from direct application of existing text-processing tools. In particular, the text metadata usually comes as a list of keywords, whereas most of the current text processing tools focus on coherent text documents. The text annotations of images further demonstrate higher level of noise and a different statistics of term usage. Therefore, we decided to create our own specialized tools for data cleaning, optimized for the particular knowledge base we employ.

The Profiset collection cleaning and disambiguation were two objectives of a bachelor thesis [26], which was successfully defended in 2012. The data cleaning comprised the removal of stopwords, translation of non-English words, and spelling corrections. The cleaned metadata contain only words that were found in a standard English dictionary, WordNet database, or among Wikipedia entries. The disambiguation of keywords, e.g. the identification of the correct meaning of individual words, is a more challenging task. To be able to decide the correct meaning of a given keyword, it is necessary to study its context. This is more difficult with keyword annota-

tions than with standard text documents, as we cannot analyze the syntactic structures. However, it is possible to obtain some information from the context of the set of words that are assigned to the same image. In our approach, we study the relatedness of keywords using the WordNet hierarchy of synsets. For each image in the dataset, we take all its keywords and try to link them to the WordNet synsets in such a way that the distances between synsets assigned to a single image is minimal. The distance is measured by the number of linking steps in the WordNet hierarchy.

After the cleaning phase, we have been able to reduce the number of distinct keywords in the Profiset database from 1 171 887 to 259 076, which means considerable improvement of metadata quality and increased probability that matching words will be found in the annotation process. It was not possible to link all the keywords to WordNet synsets with acceptable precision, but we were able to determine the links for 48 % of all keywords. Combining the original, human-created text metadata with the automatic analysis of tag relatedness and identification of synsets, we created the richest annotation knowledge base we are aware of. The improvement of annotation quality can be observed in the experiments reported in Section 7.2.3.

Annotation Forming

In our baseline solution, the annotation is formed by the keywords that appear most frequently in the metadata of images produced by the image retrieval phase. A simple word-cloud approach is used, giving different weights to words from title and the description keywords. Some solutions (e.g. [108]) also engage the ranking of the images in the computation of keyword scores, thus increasing the impact of keywords of the nearest images. However, our preliminary experiments have shown that not taking the ranks into account provides a more robust solution – instead of relying on the precision of visual best-matches, we profit on the majority voting of a higher number of images.

The simple word-cloud approach has several weak aspects. Even if the similar images provided by the image retrieval phase are perfectly relevant, many keywords may not appear in the annotation because they are not matched due to typing mistakes or natural language features, which enable to express the same concept by different synonymous words, on different abstraction levels, etc. If the retrieved set of images contains some irrelevant objects, irrelevant tags are likely to be propagated to the result. Also, the vocabulary of the annotation is determined by the keywords appearing in the source dataset and can be biased by a specific community slang.

We attempt to improve the keyword matching by employing the WordNet hierarchy of concepts, which provides means to understand various relationships between natural language terms and extend the annotation vocabulary. The utilization of WordNet is often reported as tricky, as mistakes often occur in the initial choice of synsets for individual keywords and the whole processing then increases annotation noise rather than improves the result. In our refined knowledge base, however, we have already assigned the synsets to many keywords with a rather high level of precision, using the context of keywords attached to the same image. We exploit the synsets in the following way:

- first, we create a set S of synsets related to any of the input images, and compute their frequencies;
- next, we exploit selected WordNet relationships to increase the score of synsets that are related to some other concepts in S – e.g. the synset for *dog* will increase the score of *animal* synset if both are in S.

The use of synsets automatically solves the problem of matching synonymous keywords. When the processing of synsets is completed, the keywords from the highest-scoring synsets are joined with those keywords of the input images that were not linked to any synset. Depending on their scores, the synsets can be represented by one or several keywords. Top keywords in a standard frequency ranking then form the final annotation.

The boosting of scores between related synsets improves the chances of relevant keywords to appear in the result and also partly solves the problem of irrelevant words, which should not be able to gain high scores unless too many irrelevant objects are present among the similar images. Still, some irrelevant words are likely to appear in the annotation. To remove these, a more thorough analysis of relatedness between the concepts in the final annotation result will be provided in a future version of the Annotation Tool.

7.2.3 Evaluation

Experimental evaluation is an essential part of any multimedia processing tool development, as it is not possible to analyze the proposed solutions theoretically due to the high complexity of the data in question, i.e. the image content and the natural language in our application. In the experiments presented in this section, we aim at discovering the efficiency of

Knowledge base	**Image search: method**	**result size**	**Annotation forming**
baseline solution			
original Profiset data	MPEG-7 basic search	20	most frequent keywords
refinement techniques			
cleaned Profiset data	MPEG-7 basic search	10-500	synset matching, utilization of WordNet relationships
	MPEG-7 basic search clustering rank	40-200 20-50	
	MPEG-7 basic search SIFT rank	40-100 20	

Table 7.1: Annotation techniques and parameters.

the described methods and identifying relations between various parameter settings and the annotation performance. The results will enable us to optimize the MUFIN Annotation Tool and identify topics for future research.

Table 7.1 provides an overview of search methods that were tested, as well as the retrieval parameter values. Each of the refinement techniques was tested separately, as to be able to determine its performance. The following sections describe the methodology of our testing and discuss the results.

Methodology

Similar to the general image retrieval benchmarking, the evaluation of image annotation performance is a difficult task. Fortunately, the domain of possible relevant expressions is not as large as in the case of image search, but still the desired annotation depends on application context and it is not possible to create a universal annotation ground truth. The problem is often solved by limiting the annotation vocabulary, which is employed e.g. in the ImageCLEF annotation contests [128]. However, such simplification does not reflect the real needs of many applications such as the tag hinting.

Therefore, we utilized the same solution as in the case of image search evaluation (described in detail in Section 6.3.3). We asked users to sort the

results of annotation methods into several categories: *very good*, *acceptable*, *undecidable*, *bad*. The *undecidable* category was newly added since it is sometimes not possible to decide the relevance of some concepts without a more thorough knowledge of a given location, flower species, etc. Several people participated in the evaluation process, providing at least two judgements for each query object and keyword. Two hundred randomly selected images from the Profiset collection were used as the query objects. Top 20 keywords from each annotation method were put into the evaluation pool.

For the evaluation of results, we applied several measures. First, we compared the average precision achieved by each method, for which the relevance categories were transformed into relevance percentage. The next measure only counted such objects that were considered at least *acceptable* by at least one person. The most strict measure only worked with objects that were considered at least *acceptable* at least once and were never marked as *bad*. Since the results show the same tendencies under all these measures, we only report the second one which we consider most fitting for the tag hinting use case.

Discussion of Results

Table 7.2 presents the evaluation results for selected search runs that illustrate the performance of the individual techniques. Apart from the overall relevance, expressed as the average number of relevant keywords, we can also see the number of queries for which a given method provided better, or worse, results than a relevant baseline. Wall-clock time was used to measure the processing costs.

As we can see, the main improvement of the annotation precision as compared to the baseline solution was achieved by the utilization of the cleaned reference database. We have also been able to further increase the relevance of results by employing a larger number of similar images for the annotation forming. As for the other techniques we tested, their overall results are less optimistic. On the other hand, even the less successful methods have been able to increase result quality in a number of cases. Let us now take a closer look on the individual processing phases.

Text preprocessing The impact of the Profiset collection cleaning on the annotation quality is clearly visible in the experimental results. We were able to increase the number of relevant words per result by 1.58 in average, while 86 % of queries achieved same or better relevance. The evaluation costs were naturally not influenced by the cleaning. Since the utilization of

method	result size	better/worse results [#]	average relevance	time costs [s]
baseline solution (*B1*)	20	–	8,69	2,50
cleaned dataset (*B2*)	20	142/29 vs. *B1*	10,27	2,50
cleaned dataset	50	101/66 vs. *B2*	10,91	2,80
cleaned dataset	500	110/57 vs. *B2*	11,41	4,50
clustering rank	40→20	71/89 vs. *B2*	10,18	3,10
clustering rank	100→20	83/96 vs. *B2*	9,93	3,30
SIFT rank	40→20	55/118 vs. *B2*	8,93	7,90
synset matching	20	72/77 vs. *B2*	10,21	2,70

Table 7.2: Annotation experiments – evaluation of results.

the cleaned data is definitely profitable, we employ it with all other refinements and consider the top-20 MPEG-7 search over the cleaned data as a new baseline.

Retrieval of similar images In the first set of experiments, we employed the basic MPEG-7 retrieval and tested the influence of different number of similar images on the annotation quality. As depicted in Figure 7.3, we found that the precision of results continues to grow with the number of images, but the improvements are less pronounced with higher size values. On the other hand, the time costs rise linearly. Image set sizes from the interval of [50,100] seem to provide the most balanced trade-off between effectiveness and efficiency.

The results for solutions that exploit ranking techniques are more difficult to interpret. We can see in Table 7.2 that none of the presented methods achieved better average relevance than the cleaned baseline. However, the clustering rank as well as the SIFT rank are able to improve the result quality for a number of queries. When we increase the number of objects that enter the ranking, the number of improved results grows, but there are also more results where the overall relevance gets worse. This shows us that it is not possible to simply apply a generic re-ranking method to all objects. Instead, we need to analyze the annotation behavior in more detail to determine when a given ranking should be used.

Altogether, we can observe that the ranking methods are not as effective in the context of image annotation as we hoped. Evidently, the similarity

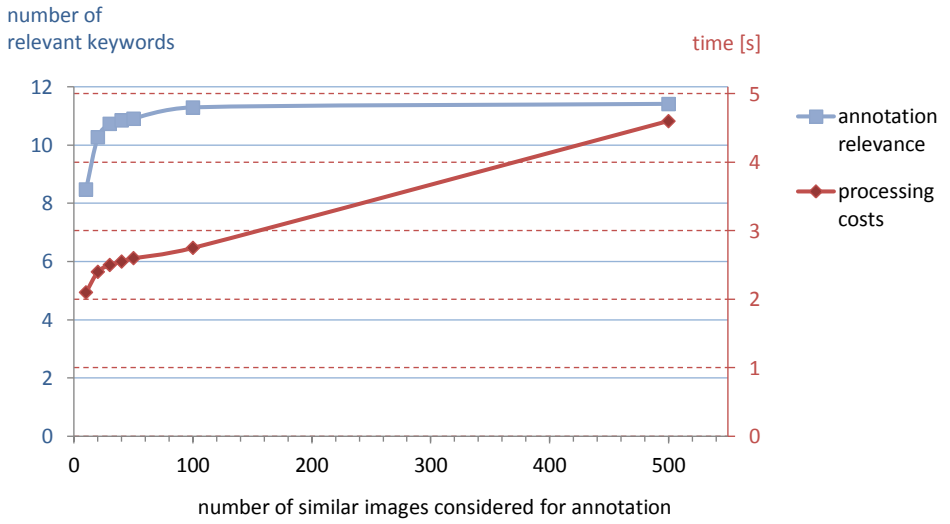

Figure 7.3: The tradeoff between result precision and evaluation costs for different number of similar images.

measures employed in the ranking procedures are not semantic enough to reliably distinguish relevant images for all types of queries. In future, we plan to try to determine the dependencies between various query properties and the suitability of ranking methods, as debated in Section 5.3. On the other hand, reasonable annotation results are obtained by processing keywords from a larger set of images, where the majority voting can balance the occasional irrelevant images. From the solutions we examined, the top-100 MPEG-7-based search is thus the most suitable option.

Annotation forming As shown in Table 7.2, the basic analysis of WordNet relationships we applied on the keywords of similar images did not succeed to improve the annotation quality noticeably. Similar to the ranking techniques, the use of WordNet was beneficial for some queries but spoiled the results in other cases. Still, we believe that the utilization of semantic relationships is the right direction for annotation improvement. However, it will be necessary to exploit the relationships more carefully to achieve better results. The bottom-up processing that we have applied increases the scores of keywords under certain conditions, but it should be complemented by a top-down checking of consistence of the final annotation. Adding such consistence metric will be our primary future task.

7.2.4 Image Annotation Software

The experimental results prove that even though there is a lot of space for improvements of the MUFIN Annotation Tool, the current implementation is capable of providing 11 relevant keywords for an image in average. Thus, it may already be interesting for people who are creating text annotations of their images and are looking for inspiration, or want to save time and effort needed for hand-typing the tags. Visually impaired people represent another group of users that might profit from the tool, as it can help them to understand images on the web that are not provided with explanatory descriptions.

All the functionality of the annotation tool is implemented within the MESSIF framework and accessible via the MUFIN Annotation Demo interface[1]. To offer potential users an easy access to the annotations, we also wrapped the MUFIN Annotation Tool into a plugin for the Mozilla Firefox web browser. The annotation can thus be obtained by two mouse-clicks for any publicly available web image. The plugin application communicates with the MUFIN search engine via a web service. Upon a request, the search engine downloads the query image, extracts its visual features, and provides the annotation keywords. Advanced users can also adjust some of the annotation parameters, e.g. the number of similar images to be used to generate the annotation. Figure 7.4 shows the output provided by the annotation plugin. The MUFIN Image Annotation plugin is now freely available from the web page `http://mufin.fi.muni.cz/plugins/annotation`.

7.3 MUFIN Image Classification

In this section, we report on our participation in the ImageCLEF 2011 Annotation Task, which is the second problem for which we tried to apply the search-based text extraction methods. Despite its name, we prefer to understand the ImageCLEF Annotation Task as a multi-label classification of images, as it required to select relevant tags from a relatively small dictionary and a training dataset was provided. In contrast to other solutions, which exploited machine learning techniques commonly used for this type of assignments, we based our approach on search-based annotation and its transformation into the classification labels. The following sections describe the task in more detail, present our solution, and discuss the competition results. More details about our solution can be found in [31].

[1] http://mufin.fi.muni.cz/annotation/

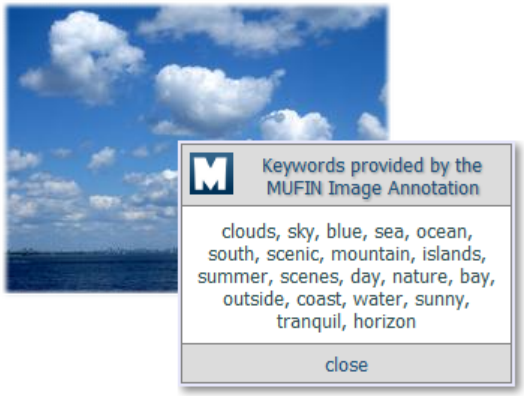

Figure 7.4: The output of the MUFIN Image Annotation plugin.

7.3.1 Task Description

In the ImageCLEF Photo Annotation Task, the participants were asked to assign relevant keywords to 10,000 test images. Part of the test objects contained only an image, for other some EXIF metadata or uncleaned Flickr tags were also available. The keywords to be assigned were chosen from a fixed set of 99 concepts, which described the scene (indoor, outdoor, landscape, ...), depicted objects (car, animal, person, ...), the representation of image content (portrait, graffiti, art, ...), events (travel, work, ...), quality issues (overexposed, blurry, ...), and sentiment concepts (happy, scary, melancholic, ...). A set of 8,000 labeled training images was available as well as a simple ontology providing relationships between the concepts. The full setup of the contest is described in the Task overview paper [128].

Unfortunately, the quality of the training data was not very high. This was caused by the fact that part of the training data was obtained in a crowdsourcing way, using workers from the Amazon Mechanical Turk portal. Even though the organizers of the contest did their best to ensure that only sane results would be accepted, the gathered data still contained a significant amount of errors. Naturally, this limited the performance of all solutions including ours, as we also used the training data to tune the parameters of our system.

7.3.2 Our Solution

Our solution was based on an early version of the MUFIN Annotation Tool, which was described at the beginning of Section 7.2.2. The annotation key-

words were selected by a simple frequency analysis of the terms in descriptions of similar images.

The fundamental difference in the basic paradigms of MUFIN Annotation Tool and the Annotation Task is that our system provides *annotations* while the task asks for *classification*. Our system provides free-text annotation of images, using any keywords that seem relevant using the content-based searching. To be able to use our tool for the task, we needed to transform the provided keywords into the restricted set of concepts defined for the task. Moreover, even though the MUFIN tool is quite good at describing the image content it does not give much information about emotions and technical-related concepts. Therefore, we had to extend our system with new components that provide specialized processing of these concepts. The overall architecture of the processing engaged in the Annotation Task is depicted in Figure 7.5. In the following sections, we will describe the individual components in more detail.

Annotation To Concept Transformation

For most concepts from the given vocabulary, we decided about their relevance by obtaining a free keyword annotation from the MUFIN Annotation Tool and matching it to the concepts. We also utilized the provided concept ontology to eliminate inconsistencies in the classification results. This process is depicted in the left part of Figure 7.5.

The first version of the MUFIN Annotation Tool, which was used in the contest, employed a standard combination of five MPEG-7 global descriptors used by the MUFIN Image Search engine [16] to evaluate visual similarity of images. Since some textual information in form of keywords and EXIF metadata was also available for some of the test images, we utilized a combination of visual and text search where applicable. The Profiset collection, which was introduced in Section 6.3.1, was used as the knowledge base. At that time, only the original uncleaned data were available. Having obtained a set of images similar to the query object, the MUFIN Annotation Tool analyzed their text metadata and produced the most frequent keywords as the annotation.

To transform the free-text annotation into the ImageCLEF concepts, it was necessary to find the semantic relations between the individual keywords and the ImageCLEF concepts. For this purpose, we used the WordNet lexical database, which provides structured semantic information for English nouns, verbs, adjectives, and adverbs. The individual words are grouped into sets of cognitive synonyms (called synsets), which are inter-

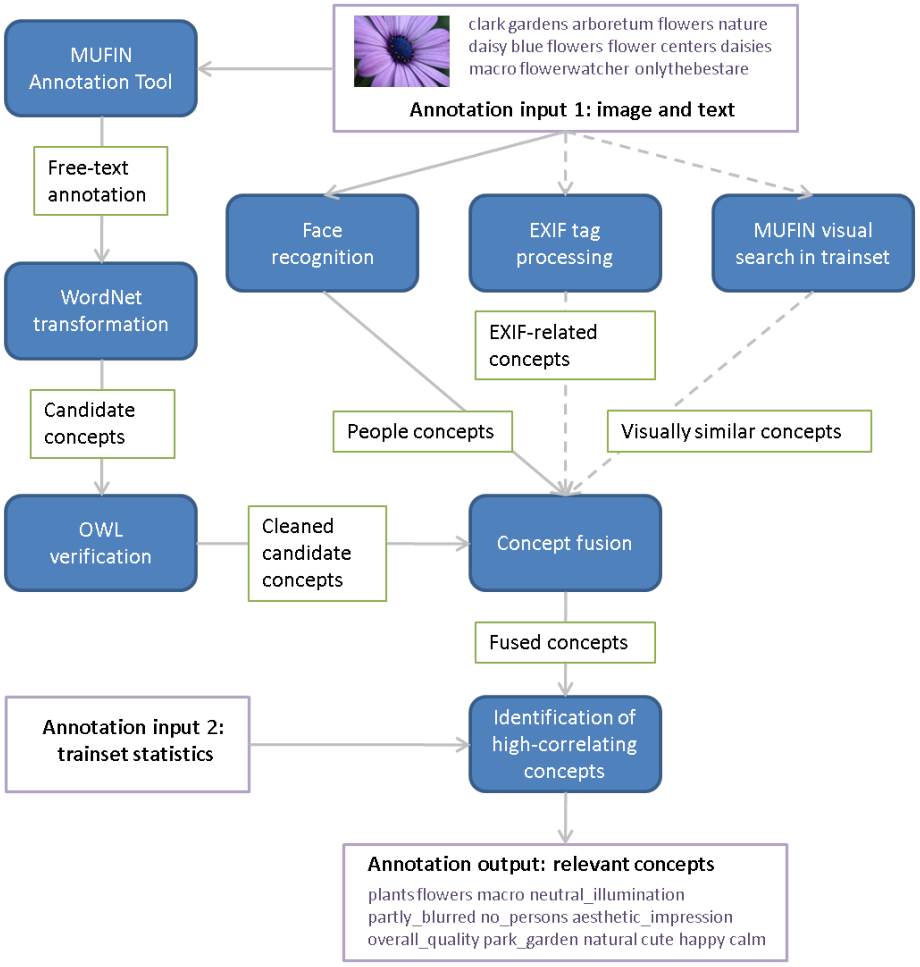

Figure 7.5: Concept retrieval schema.

linked by different semantic relations, such as hypernym/hyponym, synonym, meronym, etc. To be able to use the WordNet hierarchy to find relationships between the annotation keywords and the ImageCLEF concepts, we first needed to link both the concepts and the annotation keywords to the relevant synsets. In case of concepts, we employed a precise manual matching, whereas automatic assignment of synsets was used for the keywords. Training data provided by contest organizers was then utilized for analysis of different types of WordNet relationships and identification of those that were most successful in linking the keywords and concepts.

Using the selected synsets and relationships, we counted a relevance

score of each ImageCLEF concept during the processing of each image. The score was increased each time a keyword-synset was found to be related to a concept-synset. The increase was proportional to the confidence score of the keyword as produced by the MUFIN Annotation Tool. Finally, the concepts were checked against the OWL ontology provided within the Annotation Task. The concepts were visited in a decreasing order of their scores and whenever a conflict between two concepts is detected, the concept with the lower score was discarded.

Additional Image Processing

The mining in keywords of similar images allows us to retrieve such information as is usually contained in the image descriptions. This is most often related to image content, so the concepts related to nature, buildings, vehicles, etc. can be identified quite well. However, the Annotation Task considered also concepts that are less often described in the text, such as the number of people in the image, some background objects (*sky*, *clouds*), or image quality (*underexposed*, *blurred*). To get some more information about these, we employed the following three specialized extraction techniques: (1) a standard face recognition algorithm was applied to determine the presence of persons in the image; (2) selected EXIF tags were checked to decide about the daytime or season; and (3) MUFIN visual search over the training dataset allowed us to explore the correctly labeled data for similar visual patterns.

Definitely the most difficult concepts to assign were the ones related to user's emotions and also the abstract concepts such as *technical*, *overall_quality*, etc. By an analysis of the training dataset, we found out that even the people who annotated the trainset had problems deciding what these concepts precisely mean. Therefore, it was very difficult to determine their relevance using the image visual content. The text provided with the images was also not helpful in most cases. We finally decided to rely on the correlations between image content and the emotions it most probably evokes. For example, images of babies or nature are usually deemed cute. A set of such correlation rules was derived from the trainset and used to choose the emotional and abstract concepts.

7.3.3 Discussion of Results

As detailed in [128], three quality metrics were evaluated to compare the submitted results: Mean interpolated Average Precision (MAP), F-measure

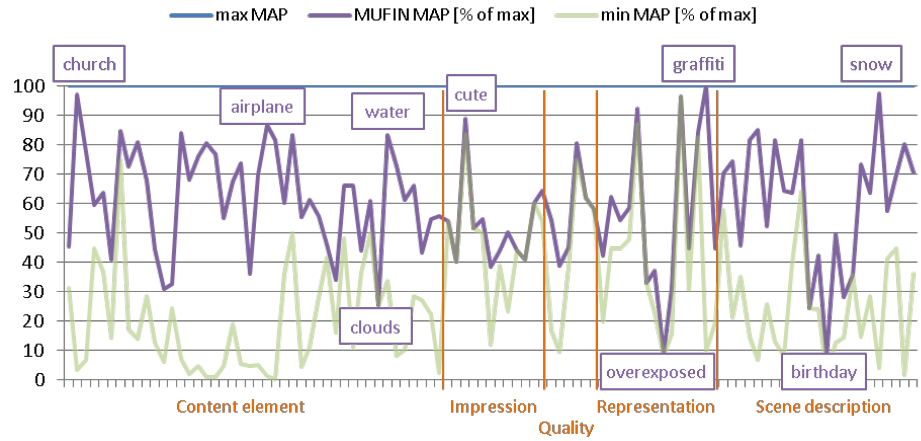

Figure 7.6: MUFIN relative MAP performance per concept.

(F-ex), and Semantic R-Precision (SR-Precision). The best of our four submissions achieved 0.299 MAP, 0.462 F-ex, and 0.628 SR-precision. After the task evaluation and the release of the algorithm for computing the MAP metric, we also re-evaluated our system with better settings of the MUFIN Annotation Tool that we have improved since the ImageCLEF task submission. Using these, we were able to gain one or two percent increase of the MAP score. With respect to the MAP measure, our solution ranked at position 13 among the 18 participating groups.

Apart from the overall results, it is also interesting to take a closer look at the performance of the various solutions for individual concepts. In particular, we are interested in the categories and particular examples of concepts where MUFIN annotation performed either well or poorly in comparison with other approaches (the absolute precision values are not so important because of varying difficulty levels of different concepts). Table 7.3 and Figure 7.6 present the results of our most successful run expressed as a percentage of the best MAP achieved for the given concept.

Table 7.3 shows the average precision of results in groups of semantically close categories as specified by the ontology provided for ImageCLEF. We can observe that the MUFIN approach is most successful in categories that are (1) related to visual image content rather than higher semantics, and (2) probable to be reflected in image tags. These are, in particular, the categories describing the depicted elements, landscape, seasons, etc. Categories related to impressions or events represent the other end of the spectrum; they are difficult to decide using only the visual information and (especially the impressions) are rarely described via tags.

Content element	Landscape elements	58.2 %	62.4 %
	Pictured objects	63.0 %	
	Urban elements	73.3 %	
Impression	Expressed impression	49.5 %	52.6 %
	Felt impression	58.6 %	
Quality	Aesthetics	60.0 %	56.3 %
	Blurring	54.6 %	
Representation	Art	58.6 %	56.4 %
	Impression	55.1 %	
	Macro	54.4 %	
	Portrait	62.1 %	
	Still life	42.0 %	
Scene description	Abstract categories	64.0 %	60.8 %
	Activity	49.4 %	
	Events	24.7 %	
	Place	72.5 %	
	Seasons	69.4 %	
	Time of day	67.9 %	

Table 7.3: MUFIN average relative MAP performance per category.

However, the average MAP values do not differ that much between categories. The reason for this is revealed if we take a closer look at the results for individual concepts, as depicted in Figure 7.6. Here we can notice low peaks in otherwise well performing categories and vice versa. For instance, the *clouds* concept in the *landscapes* category performs rather poorly. This is caused by the fact that clouds appear in many images but only as a part of a background, which is not important enough to appear in the annotation. On the contrary, airplanes are more interesting and thus mostly mentioned in the annotations. We encounter here the difference between the annotation and classification tasks – in annotation we are usually interested in the most important tags while in classification all relevant tags are wanted.

Overall, the results show that the annotation-based approach to a classi-

fication task is possible, though not as precise as the machine learning techniques. Still, our results are comparable to those of the average classifiers employed in the contest. The main advantage of our solution lies in the fact that it requires minimum training (and is therefore less dependent on the availability of high-quality training data) and is scalable to any number of concepts. For further improvements, we believe it would be useful to identify the concepts that are difficult to determine in the data-driven way and build dedicated classifiers for these. The combination of the data-driven and model-driven solutions could bring significant improvements.

7.4 Summary

Automatic image annotation is a highly desirable application of content-based image retrieval which has a direct use in many situations, such as web gallery image tagging, keyword image search, etc. In this section, we have presented the MUFIN Annotation Tool – a software designed to provide keyword annotation for arbitrary web images exploiting the data-driven paradigm. We have demonstrated that the search-based annotation is capable of providing promising results both in the context of image classification and free text annotation, and we have developed some novel techniques that are capable of improving the annotation performance. The first results of our work in this area were reported in [31, 32]. At the moment, the annotation functionality can be easily accessed by web users via a web browser plugin. In the future, we would like to integrate the MUFIN Annotation Tool into the Profimedia image stock management interface to further increase its utilization.

7.4.1 Future Research Directions

The research directions that are open in the field of data-driven image annotation are manifold. For the near future, we would like to focus on the following topics:

- Search for other knowledge sources: The data-driven approach is similar to the crowdsourcing philosophy – the more similar images are found and explored, the higher is the chance of retrieving a correct and rich annotation. The Profiset collection has proved to be a useful source of information, but there are still not enough similar images for many concepts. Therefore, we need to look for other image sets. The collection of 80 million tiny images presented in [150] is one of the

candidates. When more useful knowledge bases become available, we shall also resume the work on combining multiple information sources discussed at the beginning of Section 7.2.2.

- Improvement of image retrieval by dedicated classifiers: As we have mentioned, specialized image classifiers can be trained to identify a limited set of concepts quite precisely. Face recognition applied by many web galleries is the most common application of this principle. The integration of classifiers for selected frequent concepts will increase the precision of the annotation.

- More advanced utilization of ontologies for identification of relevant and irrelevant concepts: As we could see, the current implementation does not exploit the available WordNet hierarchy in all processing steps – in particular the final annotation cleaning could be improved. We are also interested in finding and utilizing other ontologies and semantic knowledge sources.

- Hierarchic annotation: In a longer perspective, we believe that an iterative approach to annotation retrieval will allow a significant improvement of annotation results. When a suitable hierarchy of visual concepts is established, the annotation process should first determine the relevant basic-level concepts and then refine the detailed descriptions of the concept [152]. We intend to study this annotation model with the use of the LSCOM ontology top concepts, which seem to be relatively well suited for the image annotation even though they were originally proposed for video news.

- Evaluation: The experiments presented in this chapter only compare various implementations of the MUFIN Annotation Tool. Naturally, we are also interested in an evaluation of performance in the context of other general-purpose annotation systems, such as ALIPR. To obtain a larger and more representative set of evaluations, we would also like to add some user-feedback functionality to the annotation plugin.

Chapter 8

Conclusions and Challenges

Efficient and effective retrieval of multimedia and other complex data is indisputably becoming a necessity in the modern information society. By its nature, such data requires different management techniques than those offered by traditional database or text-retrieval engines. The search paradigms are shifting from exact match to similarity-based, from precise retrieval to approximate searching, and from fixed quality measures to user-perceived relevance and flexible searching. At the same time, the scalability issues acquire new dimensions. The retrieval task has become very complex and challenging, and additional questions continue to appear with new data types and applications. On the other hand, the enormous amounts of digital information that are available nowadays enable us to perform information mining in an unprecedented extent. As debated in [115], the desired semantic information retrieval is yet far ahead of our current technology. However, state-of-the-art research already offers many interesting retrieval tools as well as theoretical pieces of knowledge that allow us to identify and pursue promising research directions.

In this work, we have studied techniques and elaborated on challenges of large-scale image retrieval, focusing in particular on web-like searching. This task has its specific requirements, the most important of which are flexibility, efficiency, and scalability. In the course of this work, we have examined the possible means of achieving these qualities, developed methods of evaluating the retrieval performance, and explored possibilities of applying the image search methods to related tasks.

The first part of the thesis has introduced the image retrieval field and provided the necessary background for our research. At the beginning, we have presented an overview of the principal open problems of image retrieval and identified our main objectives. In Chapters 3 and 4, we have proceeded by a survey of state-of-the-art image search techniques, which has introduced the concept of mono-modal and multi-modal retrieval. In Chapter 3, we have analyzed three fundamental approaches to mono-modal retrieval and demonstrated their strengths and weaknesses in the context of

real-world image search applications. The survey has been extended in Chapter 4, where we have first clarified how the combination of multiple modalities can improve both efficiency and effectiveness of complex data retrieval. Then, we have presented a novel comprehensive classification of existing multi-modal retrieval techniques, in which we have studied several aspects that are highly important for large-scale searching. We have concluded the survey part of the thesis by an overview of selected open problems in the field of big data management.

In the central part of this work (Chapter 5), we have presented our contributions in the field of the multi-modal image retrieval. First, we have focused on the development of the MUFIN search system, which has been extended with new symmetric and asymmetric late fusion algorithms. Even though we have only considered the text and visual modalities in the presented solutions, the proposed methods are flexible and applicable to many other features. We have subsequently utilized the extended MUFIN implementation in a comprehensive evaluation of state-of-the-art late fusion techniques for image search. We have compared the objective and user-perceived performance of individual methods and analyzed the dependencies between query object properties, target database characteristics, and suitability of specific retrieval techniques. Our experience with multi-modal searching has then been utilized in a proposal of an extension to the SQL language. The SimSeQL language provides novel constructs for similarity-based searching and allows users to issue multi-modal queries.

The final two chapters have been devoted to the general issues of large-scale retrieval evaluation and two practical applications of image retrieval methods. In Chapter 6, we have described the development of a new evaluation platform which allows researchers to perform a fair comparison of image retrieval techniques in a real-world environment. Chapter 7 has been dedicated to the applications, focusing in particular on image annotation and classification tasks. We have discussed the applicability of general-purpose retrieval for these tasks and presented a general model of search-based annotation. Then, we have described the architecture of the MUFIN Annotation Tool and demonstrated its usefulness in two real-world tasks.

The results of the research activities discussed in this thesis have been presented in one international journal publication, 7 full papers at international conferences and workshops, and 2 demonstration papers (a more detailed overview of our publications can be found in Appendix A). Another journal paper is currently being prepared for publication. Furthermore, we have created one software product (the MUFIN Annotation Tool) and extended the MESSIF library of functions for similarity-based retrieval.

Future Work

In several parts of this thesis, we have mentioned possible topics and directions of future research. Considering the extent of the retrieval task and the variability of its possible applications, it is obvious that the number of open problems is very large. In this section, we outline three areas in which we would like to continue our research, and identify the most immediate challenges within each of these.

In the first place, we would like to further develop our analysis of dependencies between various query and dataset characteristics and the suitability of available methods. We intend to focus on the categories of queries identified in our previous experiments, study their behavior in more detail, and identify relevant indicators that determine the suitability of specific search methods. To achieve this, it will be necessary to analyze more closely different types of information available during query processing, model their relationships, and evaluate additional experiments that will provide data for a thorough statistical analysis. Eventually, this research should result in a proposal of a heuristic strategy that would be able to recommend a suitable evaluation method for a particular query. Such heuristics would be a very useful part of any query optimization technique.

Our second suggestion concerns the similarity search language SimSeQL. Having laid its formal foundations, we plan to proceed with the research of query optimization strategies that would utilize the reformulation capabilities of the language. We would also like to create an intuitive (graphical) query formulation tool and, possibly, a conversion mechanism into the MPEG7 Query Format for inter-system communication.

Finally, a great many possibilities and challenges are connected to the image annotation and classification tasks. Having established the basic annotation functionality within the MUFIN system, we would like to further extend it and develop working tools for real-world (commercial) applications. Apart from the research and development of search-based annotation as discussed in Section 7.4.1, we would also like to build a general annotation/classification framework within MUFIN that would allow integration of the search-based approach with other information-mining techniques. In particular, we believe that significant improvements can be achieved by a combination of the search-based annotation, machine learning, and interactive data-mining with user-provided relevance feedback.

Appendix A

List of Author's Publications

This chapter reviews the research papers written by Petra Budíková (née Kohoutková) and clarifies the author's contribution to each of these works.

Journal Papers

1. P. Budikova, M. Batko, D. Novak, and P. Zezula. Large-scale multi-modal image search: theory and practice. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, 2013. Accepted for publication.

   [Author's contribution: 50 % (survey and classification of fusion techniques, comprehensive evaluation of approaches)]

Conference Papers

1. P. Budikova, M. Batko, and P. Zezula. Query language for complex similarity queries. In *6th East European Conference on Advances in Databases and Information Systems (ADBIS 2012)*, pages 85–98, 2012.

   [Author's contribution: 70 % (query language analysis and design)]

2. P. Budikova, M. Batko, and P. Zezula. Multi-modal image search for large-scale applications. In *International Workshop on Multimedia Databases and Data Engineering (MDDE 2012)*, pages 1–7, 2012.

   [Author's contribution: 70 % (survey of techniques, experimental setup and evaluation)]

3. P. Budikova, M. Batko, and P. Zezula. Evaluation platform for content-based image retrieval systems. In *International Conference on Theory and Practice of Digital Libraries (TPDL 2011)*, pages 130–142, 2011.

   [Author's contribution: 60 % (evaluation platform design and creation)]

4. P. Budikova, M. Batko, and P. Zezula. MUFIN at ImageCLEF 2011: Success or Failure? In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

   [Author's contribution: 50 % (design and implementation of the annotation and classification tools)]

5. P. Budikova, M. Batko, and P. Zezula. Similarity query postprocessing by ranking. In *8th International Workshop on Adaptive Multimedia Retrieval – Revised Selected Papers*, volume 6817 of LNCS, pages 159–173, 2011.

   [Author's contribution: 50 % (design and implementation of ranking methods)]

6. M. Batko, P. Kohoutkova, and D. Novak. CoPhIR image collection under the microscope. In *2nd International Workshop on Similarity Search and Applications (SISAP 2009)*, pages 47–54, 2009.

   [Author's contribution: 30 % (normalization and aggregation of monomodal distances)]

7. M. Batko, P. Kohoutkova, and P. Zezula. Combining metric features in large collections. In *24th International Conference on Data Engineering Workshops (ICDE 2008)*, pages 370–377, 2008.

   [Author's contribution: 30 % (analysis of TA behavior)]

Demonstration Papers

1. P. Budikova, M. Batko, and P. Zezula. Online image annotation. In *4th International Conference on Similarity Search and Applications (SISAP 2011)*, pages 109–110, 2011.

   [Author's contribution: 70 % (annotation tool design and implementation)]

2. P. Budikova, M. Batko, and P. Zezula. Improving the image retrieval system by ranking. In *3rd International Workshop on Similarity Search and Applications (SISAP 2010)*, pages 123–124, 2010.

   [Author's contribution: 70 % (demonstration design and implementation)]

Software

1. P. Budikova. *MUFIN Image Annotation*. Software tool for automatic image annotation. 2012. `http://mufin.fi.muni.cz/plugins/annotation`.

   [Author's contribution: 100 %]

Technical Reports

1. P. Budikova, M. Batko, and P. Zezula. Query language for complex similarity queries. In *Computing Research Repository (CoRR)*, pages 1–22, 2012.

   [Author's contribution: 70 % (query language analysis and design)]

Bibliography

[1] Y. Alemu, J. bin Koh, M. Ikram, and D.-K. Kim. Image retrieval in multimedia databases: A survey. In *5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2009)*, pages 681–689. IEEE Computer Society, 2009.

[2] S. Alsubaiee, A. Behm, and C. Li. Supporting location-based approximate-keyword queries. In *18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM-GIS 2010)*, pages 61–70, 2010.

[3] G. Amato, F. Falchi, C. Gennaro, F. Rabitti, and P. Savino. Improving image similarity search effectiveness in a multimedia content management system. In *Proceedings of International Workshop on Multimedia Information Systems (MIS 2004)*, pages 139–146, 2004.

[4] G. Amato, G. Mainetto, and P. Savino. A query language for similarity-based retrieval of multimedia data. In *First East-European Symposium on Advances in Databases and Information Systems (ADBIS'97)*, pages 196–203, 1997.

[5] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[6] F. S. P. Andrade, J. Almeida, H. Pedrini, and R. da Silva Torres. Fusion of local and global descriptors for content-based image and video retrieval. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP 2012)*, pages 845–853, 2012.

[7] P. André, E. Cutrell, D. S. Tan, and G. Smith. Designing novel image search interfaces by understanding unique characteristics and usage. In *12th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT '09)*, pages 340–353. Springer-Verlag, 2009.

[8] A. Arampatzis, J. Kamps, and S. Robertson. Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In *32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 524–531, 2009.

[9] A. Arampatzis, K. Zagoris, and S. A. Chatzichristofis. Dynamic two-stage image retrieval from large multimodal databases. In *33rd European Conference on IR Research (ECIR)*, volume 6611 of *LNCS*, pages 326–337, 2011.

[10] J. Aspnes and G. Shah. Skip graphs. *ACM Transactions on Algorithms*, 3(4), 2007.

[11] P. K. Atrey, M. A. Hossain, A. El-Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010.

[12] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[13] M. C. N. Barioni, H. L. Razente, A. J. M. Traina, and C. Traina Jr. Seamlessly integrating similarity queries in SQL. *Software - Practice and Experience*, 39(4):355–384, 2009.

[14] J. M. Barrios and B. Bustos. Automatic weight selection for multi-metric distances. In *Proceedings of the 4th International Conference on Similarity Search and Applications (SISAP 2011)*, pages 61–68. ACM Press, 2011.

[15] M. Batko, V. Dohnal, D. Novak, and J. Sedmidubský. MUFIN: A Multi-feature Indexing Network. In *Second International Workshop on Similarity Search and Applications (SISAP 2009)*, pages 158–159, 2009.

[16] M. Batko, F. Falchi, C. Lucchese, D. Novak, R. Perego, F. Rabitti, J. Sedmidubsky, and P. Zezula. Building a web-scale image similarity search system. *Multimedia Tools and Applications*, 47:599–629, 2010.

[17] M. Batko, P. Kohoutkova, and D. Novak. CoPhIR image collection under the microscope. *2nd International Workshop on Similarity Search and Applications (SISAP 2009)*, pages 47–54, 2009.

[18] M. Batko, P. Kohoutkova, and P. Zezula. Combining metric features in large collections. In *24th International Conference on Data Engineering Workshops (ICDE 2008)*, pages 370–377, 2008.

[19] M. Batko, D. Novak, F. Falchi, and P. Zezula. On scalability of the similarity search in the world of peers. In *Proceedings of the 1st international conference on Scalable information systems (InfoScale '06)*. ACM, 2006.

[20] M. Batko, D. Novak, F. Falchi, and P. Zezula. On scalability of the similarity search in the world of peers. In *1st International Conference on Scalable Information Systems (Infoscale 2006)*, page 20. ACM, 2006.

[21] M. Batko, D. Novak, and P. Zezula. MESSIF: Metric similarity search implementation framework. In *1st DELOS Conference*, volume 4877 of *LNCS*, pages 1–10. Springer, 2007.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[23] T. Berber and A. Alpkocak. An extended vector space model for content based image retrieval. In *10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009)*, LNCS, pages 219–222, 2009.

[24] P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2, 2009.

[25] É. Bossé, J. Roy, and S. Wark. *Concepts, models, and tools for information fusion*. Artech House intelligence and information operations library. Artech House, Inc., 2007.

[26] J. Botorek. Processing tool for multimedia data annotations. Bachelor thesis, Masaryk University, Faculty of Informatics, 2012.

[27] A. Bozzon and P. Fraternali. Chapter 8: Multimedia and multimodal information retrieval. In *Search Computing*, volume 5950 of *LNCS*, pages 135–155. Springer Berlin / Heidelberg, 2010.

[28] P. Budikova, M. Batko, D. Novak, and P. Zezula. Large-scale multimodal image search: theory and practice. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, 2013. Accepted for publication.

[29] P. Budikova, M. Batko, and P. Zezula. Improving the image retrieval system by ranking. In *3rd International Workshop on Similarity Search and Applications (SISAP 2010)*, pages 123–124, 2010.

[30] P. Budikova, M. Batko, and P. Zezula. Evaluation platform for content-based image retrieval systems. In *International Conference on Theory and Practice of Digital Libraries (TPDL 2011)*, pages 130–142, 2011.

[31] P. Budikova, M. Batko, and P. Zezula. MUFIN at ImageCLEF 2011: Success or Failure? In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[32] P. Budikova, M. Batko, and P. Zezula. Online image annotation. In *4th International Conference on Similarity Search and Applications (SISAP 2011)*, pages 109–110, 2011.

[33] P. Budikova, M. Batko, and P. Zezula. Similarity query postprocessing by ranking. In *8th International Workshop on Adaptive Multimedia Retrieval – Revised Selected Papers*, volume 6817 of *LNCS*, pages 159–173, 2011.

[34] P. Budikova, M. Batko, and P. Zezula. Multi-modal image search for large-scale applications. In *International Workshop on Multimedia Databases and Data Engineering (MDDE 2012)*, pages 1–7, 2012.

[35] P. Budikova, M. Batko, and P. Zezula. Query Language for Complex Similarity Queries. *Computing Research Repository (CoRR)*, pages 1–22, 2012.

[36] P. Budikova, M. Batko, and P. Zezula. Query language for complex similarity queries. In *6th East European Conference on Advances in Databases and Information Systems (ADBIS 2012)*, pages 85–98, 2012.

[37] B. Bustos, S. Kreft, and T. Skopal. Adapting metric indexes for searching in multi-metric spaces. *Multimedia Tools and Applications*, 58(3):467–496, 2012.

[38] B. Bustos and T. Skopal. Dynamic similarity search in multi-metric spaces. In *8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR 2006)*, pages 137–146, 2006.

[39] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys*, 44(1):1, 2012.

[40] S. A. Chatzichristofis, K. Zagoris, Y. S. Boutalis, and A. Arampatzis. A fuzzy rank-based late fusion method for image retrieval. In *18th International Conference on Advances in Multimedia Modeling (MMM 2012)*, pages 463–472, 2012.

[41] J. Chen, R. Ma, and Z. Su. Weighting visual features with pseudo relevance feedback for cbir. In *9th ACM International Conference on Image and Video Retrieval (CIVR 2010)*, pages 220–227, 2010.

[42] J. Chen, Y. Zhu, H. Wang, W. Jin, and Y. Yu. Effective and efficient multi-facet web image annotation. *Journal of Computer Science and Technology*, 27(3):541–553, 2012.

[43] Y. Chen, N. Yu, B. Luo, and X. wen Chen. iLike: integrating visual and textual features for vertical search. In *18th International Conference on Multimedia (ACM Multimedia 2010)*, pages 221–230, 2010.

[44] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the 8th ACM International Conference on Image and Video Retrieval (CIVR 2009)*. ACM, 2009.

[45] E. Chvez, G. Navarro, R. Baeza-Yates, and J. L. Marroqun. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[46] P. Ciaccia and M. Patella. The M^2-tree: Processing Complex Multi-Feature Queries with Just One Index. In *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, 2000.

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

[48] S. Clinchant, J. Ah-Pine, and G. Csurka. Semantic combination of textual and visual information in multimedia retrieval. In *1st ACM International Conference on Multimedia Retrieval (ICMR '11)*, pages 44:1–44:8, New York, NY, USA, 2011.

[49] E. F. Codd. *The relational model for database management: version 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[50] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[51] G. Csurka and S. Clinchant. An empirical study of fusion operators for multimodal image retrieval. In *10th International Workshop on Content-Based Multimedia Indexing (CBMI 2012)*, pages 1–6, 2012.

[52] M. d'Aquin and N. F. Noy. Where to publish and find ontologies? A survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:96–111, 2012.

[53] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. A survey of semantic image and video annotation tools. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, volume 6050 of *LNCS*, pages 196–239. Springer, 2011.

[54] R. Datta, W. Ge, J. Li, and J. Z. Wang. Toward bridging the annotation-retrieval gap in image search. *IEEE Multimedia*, 14:24–35, 2007.

[55] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40:5:1–5:60, 2008.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 248–255, 2009.

[57] A. Depeursinge and H. Müller. Fusion techniques for combining textual and visual information retrieval. In *ImageCLEF*, volume 32 of *The Kluwer International Series on Information Retrieval*, pages 95–114. Springer Berlin Heidelberg, 2010.

[58] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11:77–107, 2008.

[59] T. Deserno, S. Antani, and R. Long. Ontology of gaps in Content-Based image retrieval. *Journal of Digital Imaging*, 22(2):202–215, 2009.

[60] G. Ding, J. Wang, N. Xu, and L. Zhang. Automatic image annotations by mining web image data. In *International Conference on Data Mining (ICDM) Workshops*, pages 152–157. IEEE Computer Society, 2009.

[61] H. Ding, J. Liu, and H. Lu. Hierarchical clustering-based navigation of image search results. In *16th ACM international conference on Multimedia (ACM Multimedia '08)*, pages 741–744. ACM, 2008.

[62] M. Döller, R. Tous, M. Gruhne, K. Yoon, M. Sano, and I. S. Burnett. The MPEG Query Format: Unifying access to multimedia retrieval systems. *IEEE MultiMedia*, 15(4):82–95, 2008.

[63] J. P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35(1):3–14, 2002.

[64] H. J. Escalante, C. A. Hernández, L. E. Sucar, and M. M. y Gómez. Late fusion of heterogeneous methods for multimedia image retrieval. In *1st ACM SIGMM International Conference on Multimedia Information Retrieval (MIR 2008)*, pages 172–179, 2008.

[65] H. J. Escalante, M. M. y Gómez, and L. E. Sucar. Multimodal indexing based on semantic cohesion for image retrieval. *Information Retrieval*, 15(1):1–32, 2012.

[66] R. Fagin. Combining fuzzy information: an overview. *SIGMOD Record*, 31:109–118, 2002.

[67] F. Falchi, C. Gennaro, F. Rabitti, G. Amato, P. Savino, and P. Stankev. Improving image similarity search effectiveness in a multimedia content management system. In *10th Workshop on Multimedia Information Systems (MIS 2004)*, pages 139–146, 2004.

[68] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1778–1785, 2009.

[69] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[70] M. Ferecatu, N. Boujemaa, and M. Crucianu. Semantic interactive image retrieval combining visual and conceptual content description. *Multimedia Systems*, 13(5-6):309–322, 2008.

[71] M. Ferecatu and H. Sahbi. Telecom paristech at imageclefphoto 2008: Bi-modal text and image retrieval with diversity enhancement. In *Working Notes for the CLEF 2008 workshop*, 2008.

[72] C. Fluhr, P.-A. Moëllic, and P. Hède. Usage-oriented multimedia information retrieval technological evaluation. In *Multimedia Information Retrieval*, pages 301–306, 2006.

[73] L. Gao, M. Wang, X. S. Wang, and S. Padmanabhan. Expressing and optimizing similarity-based queries in SQL. In *23rd International Conference on Conceptual Modeling*, pages 464–478, 2004.

[74] T. Gevers and A. W. M. Smeulders. Image search engines - an overview. Technical report, University of Amsterdam, 2003.

[75] T. Gloe and R. Böhme. The Dresden image database for benchmarking digital image forensics. *Journal of Digital Forensic Practice*, 3(2-4):150–159, 2010.

[76] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The IAPR TC-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, pages 13–23, 2006.

[77] D. Guliato, E. V. de Melo, R. M. Rangayyan, and R. C. Soares. POSTGRESQL-IE: An image-handling extension for PostgreSQL. *Journal of Digital Imaging*, 22(2):149–165, 2009.

[78] A. Hanbury. A survey of methods for image annotation. *Journal of Visual Languages and Computing*, 19(5):617–627, 2008.

[79] R. He, N. Xiong, L. T. Yang, and J. H. Park. Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval. *Information Fusion*, 12(3):223 – 230, 2011.

[80] R. T. Hemayati, W. Meng, and C. T. Yu. Semantic-based grouping of search engine results using wordnet. In *Advances in Data and Web Management*, pages 678–686, 2007.

[81] T. Homola, V. Dohnal, and P. Zezula. Searching for sub-images using sequence alignment. *IEEE International Symposium on Multimedia*, 0:61–68, 2011.

[82] E. Hoque, G. Strong, O. Hoeber, and M. Gong. Conceptual query expansion and visual search results exploration for web image retrieval. In *7th Atlantic Web Intelligence Conference (AWIC 2011)*, pages 73–82, 2011.

[83] E. Hörster, M. Slaney, M. Ranzato, and K. Weinberger. Unsupervised image ranking. In *1st ACM workshop on Large-scale multimedia retrieval and mining (LS-MMRM '09)*, pages 81–88, New York, NY, USA, 2009.

[84] J. Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Business, 2008.

[85] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Reranking methods for visual search. *IEEE MultiMedia*, 14(3):14–22, 2007.

[86] X. Huang, S.-C. Chen, M.-L. Shyu, and C. Zhang. Mining high-level user concepts with multiple instance learning and relevance feedback for content-based image retrieval. In *Revised Papers from MDM/KDD and PAKDD/KDMCD*, volume 2797 of *LNCS*, pages 50–67. Springer, 2002.

[87] M. J. Huiskes and M. S. Lew. The MIR Flickr retrieval evaluation. In *Proceedings of the Multimedia Information Retrieval*. ACM, 2008.

[88] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th annual ACM symposium on Theory of computing (STOC '98)*, pages 604–613. ACM, 1998.

[89] R. Jain and P. Sinha. Content without context is meaningless. In *International conference on Multimedia (ACM Multimedia 2010)*, pages 1259–1268. ACM, 2010.

[90] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[91] H. Jegou, C. Schmid, H. Harzallah, and J. J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11, 2010.

[92] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & WordNet. In *International conference on Multimedia (ACM Multimedia 2005)*, pages 706–715. ACM, 2005.

[93] Y. Jing and S. Baluja. VisualRank: Applying PageRank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1877–1890, 2008.

[94] M. L. Kherfi, D. Ziou, and A. Bernardi. Image retrieval from the World Wide Web: Issues, techniques, and systems. *ACM Computing Surveys*, 36:35–67, 2004.

[95] J. Kludas, E. Bruno, and S. Marchand-Maillet. Information fusion in multimedia information retrieval. In *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics*, volume 4918 of *LNCS*, pages 147–159. Springer Berlin / Heidelberg, 2008.

[96] M. Kyselak, D. Novak, and P. Zezula. Stabilizing the recall in similarity search. In *4th International Conference on Similarity Search and Applications (SISAP 2011)*, pages 43–49, 2011.

[97] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2:1–19, 2006.

[98] J. Li, Q. Ma, Y. Asano, and M. Yoshikawa. Re-ranking by multi-modal relevance feedback for content-based social image retrieval. In *14th Asia-Pacific Web Conference on Web Technologies and Applications (APWeb 2012)*, pages 399–410, 2012.

[99] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

[100] J. Z. Li, M. T. Özsu, D. Szafron, and V. Oria. MOQL: A multimedia object query language. In *3rd International Workshop on Multimedia Information Systems*, 1997.

[101] J. Little, A. Abrams, and R. Pless. Tools for richer crowd source image annotations. In *IEEE Workshop on Applications of Computer Vision (WACV 2012)*, pages 369–374. IEEE, 2012.

[102] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Retagging social images based on visual and semantic consistency. In *19th international conference on World wide web (WWW '10)*, pages 1149–1150. ACM, 2010.

[103] L. Liu and M. T. Özsu, editors. *Encyclopedia of Database Systems*. Springer US, 2009.

[104] Y. Liu, T. Mei, and X.-S. Hua. CrowdReranking: exploring multiple search engines for visual search reranking. In *32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 500–507, 2009.

[105] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007.

[106] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[107] J. Magalhães and S. M. Rüger. Using manual and automated annotations to search images by semantic similarity. *Multimedia Tools and Applications*, 56(1):109–129, 2012.

[108] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *10th European Conference on Computer Vision*, volume 5304 of *LNCS*, pages 316–329. Springer, 2008.

[109] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[110] S. Marchand-Maillet and M. Worring. Benchmarking image and video retrieval: an overview. In *Multimedia Information Retrieval*, pages 297–300, 2006.

[111] M. McCandless, E. Hatcher, and O. Gospodnetić. *Lucene in Action: Covers Apache Lucene V. 3. 0*. Manning Pubs Co Series. Manning, 2010.

[112] J. Melton and A. Eisenberg. SQL Multimedia and Application Packages (SQL/MM). *SIGMOD Record*, 30(4):97–102, 2001.

[113] I. Mironica, B. Ionescu, and C. Vertan. Hierarchical clustering relevance feedback for content-based image retrieval. In *10th International Workshop on Content-Based Multimedia Indexing (CBMI 2012)*, pages 1–6, 2012.

[114] D. Morrison, S. Marchand-Maillet, and E. Bruno. TagCaptcha: annotating images with CAPTCHAs. In *International conference on Multimedia (ACM Multimedia 2010)*, pages 1557–1558, 2010.

[115] P. Morville and J. Callender. *Search Patterns - Design for Discovery*. O'Reilly, 2010.

[116] MPEG-7. Multimedia content description interfaces. Part 3: Visual. ISO/IEC 15938-3:2002, 2002.

[117] H. Müller, P. Clough, T. Deselaers, and B. Caputo. *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*. Springer, 1st edition, 2010.

[118] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, and D. M. Squire. A framework for benchmarking in CBIR. *Multimedia Tools and Applications*, 21(1):55–73, 2003.

[119] M. R. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. H. Hsu, L. S. Kennedy, A. G. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[120] A. P. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *15th International conference on Multimedia (ACM Multimedia '07)*, pages 991–1000. ACM, 2007.

[121] S. A. Noah, D. A. Ali, A. C. Alhadi, and J. M. Kassim. Going beyond the surrounding text to semantically annotate and search digital images. In *2nd International Conference, ACIIDS*, volume 5990 of *LNCS*, pages 169–179. Springer, 2010.

[122] M. Nölle, M. Rubik, and A. Hanbury. Results of the MUSCLE CIS Coin Competition 2006. In *Proceedings of the Muscle CIS Coin Competition Workshop*, pages 1–5, 2006.

[123] D. Novák, M. Batko, and P. Zezula. Web-scale system for image similarity search: When the dreams are coming true. In *International Workshop on Content-Based Multimedia Indexing (CBMI 2008)*, pages 446–453, 2008.

[124] D. Novak, M. Batko, and P. Zezula. Metric index: An efficient and scalable solution for precise and approximate similarity search. *Information Systems*, 36(4):721–733, 2011.

[125] D. Novak, M. Batko, and P. Zezula. Large-scale similarity data management with distributed metric index. *Inf. Process. Manage.*, 48(5):855–872, 2012.

[126] S. Nowak and M. J. Huiskes. New strategies for image annotation: Overview of the Photo Annotation Task at ImageCLEF 2010. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[127] S. Nowak, H. M. Lukashevich, P. Dunker, and S. M. Rüger. Performance measures for multilabel evaluation: a case study in the area of image classification. In *Multimedia Information Retrieval*, pages 35–44. ACM, 2010.

[128] S. Nowak, K. Nagel, and J. Liebetrau. The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks. In *CLEF 2011 working notes*, 2011.

[129] G. Park, Y. Baek, and H.-K. Lee. Web image retrieval using majority-based ranking approach. *Multimedia Tools and Applications*, 31(2):195–219, 2006.

[130] M. Patella and P. Ciaccia. Approximate similarity search: A multi-faceted problem. *Journal of Discrete Algorithms*, 7(1):36–48, 2009.

[131] D. C. G. Pedronette and R. da S. Torres. Exploiting contextual spaces for image re-ranking and rank aggregation. In *1st ACM International Conference on Multimedia Retrieval (ICMR '11)*, pages 13:1–13:8, New York, NY, USA, 2011.

[132] T.-T. Pham, N. Maillot, J.-H. Lim, and J.-P. Chevallet. Latent semantic fusion model for image retrieval and annotation. In *Sixteenth ACM Conference on Information and Knowledge Management (CIKM 2007)*, pages 439–444, 2007.

[133] A. Popescu, T. Tsikrika, and J. Kludas. Overview of the Wikipedia Retrieval Task at ImageCLEF 2010. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[134] C. Pulla and C. V. Jawahar. Multi modal semantic indexing for image retrieval. In *9th ACM International Conference on Image and Video Retrieval*, pages 342–349. ACM, 2010.

[135] T. Quack, U. Mönich, L. Thiele, and B. S. Manjunath. Cortina: a system for large-scale, content-based web image retrieval. In *12th International conference on Multimedia (ACM Multimedia 2004)*, pages 508–511. ACM, 2004.

[136] S. Radhouani, J. Kalpathy-Cramer, S. Bedrick, B. Bakke, and W. R. Hersh. Using media fusion and domain dimensions to improve precision in medical image retrieval. In *10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009)*, pages 223–230, 2009.

[137] F. Richter, S. Romberg, E. Hörster, and R. Lienhart. Multimodal ranking for image search on community databases. In *Proceedings of the international conference on Multimedia information retrieval (MIR '10)*, pages 63–72, New York, NY, USA, 2010.

[138] L. Rokach. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53(12):4046–4072, 2009.

[139] A. Ross and A. K. Jain. Multimodal biometrics: An overview. In *Proceedings of 12th European Signal Processing Conference*, pages 1221–1224, 2004.

[140] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):644–655, 1998.

[141] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, 1999.

[142] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):754–766, 2011.

[143] H. T. Shen, S. Jiang, K.-L. Tan, Z. Huang, and X. Zhou. Speed up interactive image retrieval. *VLDB Journal*, 18(1):329–343, 2009.

[144] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts, 6th Edition*. McGraw-Hill Book Company, 2011.

[145] T. Skopal and B. Bustos. On nonmetric similarity search problems in complex domains. *ACM Computing Surveys*, 43(4):34, 2011.

[146] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349 –1380, 2000.

[147] C. Snoek, M. Worring, and A. W. M. Smeulders. Early versus late fusion in semantic video analysis. In *13th ACM International Conference on Multimedia (ACM Multimedia)*, pages 399–402, 2005.

[148] Y. Sugiyama, M. P. Kato, H. Ohshima, and K. Tanaka. Relative relevance feedback in image retrieval. In *International Conference on Multimedia and Expo (ICME 2012)*, pages 272–277, 2012.

[149] S. Tollari, M. Detyniecki, C. Marsala, A. Fakeri-Tabrizi, M.-R. Amini, and P. Gallinari. Exploiting visual concepts to improve text-based image retrieval. In *31th European Conference on IR Research (ECIR 2009)*, pages 701–705, 2009.

[150] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[151] R. d. S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informtica Terica e Aplicada*, 13(2):161–185, 2006.

[152] A.-M. Tousch, S. Herbin, and J.-Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012.

[153] R. Troncy, B. Huet, and S. Schenk, editors. *Multimedia Semantics: Metadata, Analysis and Interaction*. Wiley-Blackwell, 2011.

[154] C. Tsinaraki and S. Christodoulakis. An MPEG-7 query language and a user preference model that allow semantic retrieval and filtering of multimedia content. *Multimedia Systems*, 13(2):131–153, 2007.

[155] T. Tsirelis and A. Delopoulos. Automatic ground-truth image generation from user tags. In *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011)*, 2011.

[156] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical report, Department of Computing Science, Utrecht University, 2002.

[157] L. von Ahn and L. Dabbish. ESP: Labeling images with a computer game. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 91–98. AAAI, 2005.

[158] K. Vu, H. Cheng, and K. A. Hua. Image retrieval in multipoint queries. *International Journal of Imaging Systems and Technology*, 18(2-3):170–181, 2008.

[159] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Scalable search-based image annotation. *Multimedia Systems*, 14(4):205–220, 2008.

[160] L. Wang, L. Yang, and X. Tian. Query aware visual similarity propagation for image search reranking. In *17th International Conference on Multimedia (ACM Multimedia 2009)*, pages 725–728, 2009.

[161] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. AnnoSearch: Image auto-annotation by search. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 1483–1490. IEEE Computer Society, 2006.

[162] T. Westerveld and R. van Zwol. The INEX 2006 Multimedia Track. In *5th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 331–344, 2006.

[163] P. Wilkins, A. F. Smeaton, and P. Ferguson. Properties of optimally weighted data fusion in cbmir. In *33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 643–650, 2010.

[164] P. Zezula. Future trends in similarity searching. In *5th International Conference on Similarity Search and Applications (SISAP 2012)*, volume 7404 of *LNCS*, pages 8–24. Springer, 2012.

[165] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search - The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer, 2006.

[166] D. Zhang, M. M. Islam, and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.

[167] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *12th European Conference on Computer Vision (ECCV 2012)*, pages 660–673, 2012.

[168] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.

[169] H. Zitouni, S. G. Sevil, D. Ozkan, and P. Duygulu. Re-ranking of web image search results using a graph algorithm. In *19th International Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, 2008.