Combining Features and Semantics: Advanced Methods for Content-based Retrieval

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Johanna Vompras
aus Pyrzyce

Januar 2009

Aus dem Institut für Informatik
der Heinrich-Heine Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Stefan Conrad
Koreferent: Prof. Dr. Martin Mauve
Tag der mündlichen Prüfung: 08.07.2009

Dedicated to

Angelos

Acknowledgements

This thesis summarizes my research at the Databases and Information Systems Group of the Department of Computer Science at the Heinrich Heine University of Düsseldorf, where I have been working as a research assistant since June 2004.

I acknowledge my thesis advisor and first referee, *Prof. Dr. Stefan Conrad*, for providing a motivating atmosphere and giving constructive comments during my research time as well as on the preliminary version of this thesis. It was a great pleasure for me to pursuit research under his supervision. I also thank the second reviewer of this thesis, *Prof. Dr. Martin Mauve*, for his interest in my work and willingness to be the second referee.

My special compliments go to my former colleagues at the database group, especially to *Evgenia Altareva*, *Cristian Pérez de Laborda* and *Christopher Popfinger*, for the amicable atmosphere, the enlightening tea breaks and the numerous activities beyond university walls. Additionally, I extend my compliments to *Hieu Le Quang* and all my new colleagues *Tim Schlüter, Katrin Zaiß*, *Jiwu Zhao*, and *Ludmila Himmelspach* for their teaching support and especially *Sadet Alčić* for several implementations and his interest in my research work. Further compliments go to *Guido Königstein*, *Marga Potthoff*, and *Sabine Freese* for their continuous technical and administrative assistance.

I want to express my deepest thanks to my parents *Irena* and *Gerhard Alichniewicz* for their love and support in each life situation. Their guidance in my younger days has created essential fundamentals for both my personal development and the achieved academic level.

Above all, I am deeply grateful to my husband *Angelos Vompras* for providing me the best imaginable family circumstances for conducting my research and for giving me love, mental support, understanding, and encouragement.

Düsseldorf, Germany
January, 2009

Johanna Vompras

Abstract

In order to reduce the 'semantic gap', which is known as the mismatch between the low-level feature representation and the high-level human perception, the inclusion of semantic knowledge into advanced content-based retrieval systems has become indispensable. One approach to overcome the gap is the manual or automatic assignment of annotations for the description of multimedia objects classifying the data into semantic categories and thus facilitating textual or conceptual queries. Although the manual approach takes away the uncertainty of fully automatic annotation, but in return it requires a high effort. Hence, an interactive combination of the automatic computation and semantic modeling would provide a significant improvement by eliminating the disadvantages of the both approaches. For this purpose, we present several concepts and architectures that are specifically developed to attenuate different manifestations of the semantic gap.

At first, we introduce a framework for supporting semi-automatic annotation of multimedia data which is based on the extraction of elementary low-level features, user's relevance feedback, and the usage of ontology knowledge. Further aspects of this work include the encountered problems during the annotation process, like multiple levels of abstraction at which annotations are assigned, incompleteness of annotation data, or differing users' subjectivity. To solve these problems, we introduce the Annotation Analysis Framework which provides a graph-based representation for annotations, encoding their complex structure and making them understandable for the machine by allowing semantic inference.

In order to incorporate user diversity which might negatively influence the retrieval behavior, methods for understanding and interpreting the subjective views are needed. Based on our annotation/retrieval framework, we present the *GLENARVAN* component, which is responsible for context computation, ontology comparison, and query expansion according to users' profiles. Here, we consider two different aspects: First, user diversity is modeled as different user profiles and annotation ontologies which are brought together in order to extract contextual information and thus to attenuate users' subjectivity. The second issue is how to prevent the retrieval process to fail in the case of different views on the data collection. For this purpose, the subjective annotations are used to discover mappings between the user's and the system's conceptual model, which are subsequently applied to infer additional parameters for a user-adapted query.

Finally, we propose a Pseudo Relevance Feedback method, which improves the content-based image retrieval by query reformulation. The particular aspect of this method is the fact that the involved functions, like result judgments, relevance computation, and reordering of the results, have been implemented as user-defined functions, making the method highly suitable for web retrieval applications.

Zusammenfassung

Als 'semantische Lücke' wird der Unterschied zwischen der begrenzten Ausdruckskraft der aus Rohdaten automatisch extrahierbaren *low-level* Merkmalen und der menschlichen *high-level* Wahrnehmung von Inhalt und Ähnlichkeit bezeichnet. Um diese zu minimieren, ist das Einbringen von semantischem Wissen in moderne inhaltsbasierte (*content-based, CBIR*) Information Retrieval Systeme unbedingt notwendig. Einen weit verbreiteten Ansatz dazu stellt die inhaltliche Annotation von multimedialen Objekten dar, die diese Daten in semantische Kategorien klassifiziert und somit textuelle oder konzeptuelle Anfragen möglich macht. Obwohl der Ansatz der manuellen Annotation der mit Unsicherheiten behafteten automatischen Annotation gegenübersteht, ist dieser dafür mit einem hohen Aufwand verbunden. Die Nachteile beider Vorgehensweisen könnten jedoch durch einen interaktiven Prozess, der die automatische Berechnung und die semantische Modellierung kombiniert, eliminiert werden. Dazu präsentieren wir mehrere speziell für CBIR Systeme entwickelten Konzepte und Architekturen, um die verschiedenen Ausprägungen der semantischen Lücke abzuschwächen.

Zuerst stellen wir unser Framework für die semi-automatische Annotation von Multimedia-Daten vor, welches auf der automatischen Extraktion von low-level Merkmalen, Relevance Feedback und der Benutzung von Wissen aus Ontologien basiert. Weitere Aspekte der Arbeit behandeln die während des Annotationsprozesses auftretenden Probleme, wie die Existenz von unterschiedlichen Abstraktionsebenen, die Unvollständigkeit der Annotationsdaten oder die zwischen den Benutzern eines Systems variierende Subjektivität. Um die genannten Probleme zu lösen, wird unser System für die Analyse von Annotationen vorgestellt, welches diese in eine graph-basierte Repräsentation überführt und sie somit für den Benutzer nachvollziehbar und durch die gegebenen Inferenz-Funktionen für die Maschine verständlich macht.

Um zu vermeiden, dass eine große Benutzerdiversität das Retrievalverhalten eines IR Systems negativ beeinflusst, werden Methoden für das Verstehen und Interpretieren der subjektiven Wahrnehmung der Benutzer benötigt. Dazu wird aufbauend auf unserem Annotations/Retrieval System das *GLENARVAN* Teilsystem präsentiert, welches für die Kontextberechnung, den Vergleich von Annotationsontologien und die Anfrageerweiterung (*query expansion*) anhand von Benutzerprofilen zuständig ist. Es werden hierbei zwei Aspekte betrachtet: Zuerst wird die Benutzerdiversität durch eine Menge von Benutzerprofilen und den dazugehörigen Annotationsontologien modelliert und dafür verwendet, Kontextinformation zu extrahieren und somit die Subjektivität der Benutzer abzuschwächen. Der zweite Aspekt beschäftigt sich mit der Frage, wie man trotz verschiedener Sichten auf identische Datenbestände zufriedenstellende Retrievalergebnisse erreichen kann. Als Lösung wird hier ein Query Expansion Algorithmus vorgestellt, der anhand der subjektiven Annotationen die Zuordnungen zwischen der Systemontologie und dem vom Benutzer verwendeten Vokabulars aufdeckt und somit

zusätzliche Parameter für eine an den jeweiligen Benutzer angepasste Anfrage liefert.

Anschließend stellen wir unsere Methode des Pseudo Relevance Feedbacks für Bilddaten vor, die eine Anpassung der Anfrage (*query reformulation*) anhand der Feedbackaktivitäten des Benutzers vornimmt. Unser Verfahren eignet sich stark für die Integration in bestehende Web Retrieval Anwendungen, da die Implementierung der beinhalteten Funktionalitäten, wie der Bewertung der Ergebnisse, Relevanzberechnung und die Neuordnung der Ergebnismenge mithilfe von benutzerdefinierten Funktionen (*user-defined functions, UDF*) realisiert ist.

Contents

1

Motivation

In answer to the spread of the World Wide Web there has been an explosion in the amount of digital media, like images, videos, and audio data accessible for everyone. This development confronts researchers with new questions concerning the storage, management, and the retrieval of the large and heterogeneous repositories. The most important question for the end-user is: *"How can relevant data, which will satisfy my information need, be efficiently extracted from a flood of data?"*. To answer this question, a lot of efforts in information retrieval (IR) for textual data and content-based retrieval techniques for image and multimedia data has been done in the last decades. The commencements of IR go back to pure text retrieval, whose aim is the search for a specific piece of information (e.g. a news article) from a large document collection. In doing so, the most popular approach is to represent the documents using the vector space model, alternatively known as the bag-of-words model. Here, the basic idea is to extract n content bearing unique terms (after the foregoing elimination of stop words and application of stemming) from the union of all documents of the collection as features and then represent each document d as a vector $\vec{f}(d)$ of this n-dimensional feature space. Queries are usually formulated in natural language trying to express the user's information need. The system should then be able to transform the query into the internal representation, compare it with all document vectors in the collection, rank them according to their relevance and present the result set to the user. In contrast to image retrieval, the access to the content, and consequently the meaning of a document, is explicitly available in the terms which represent it. Thus, to some extent the semantic information can be extracted automatically from the data.

Research in content-based image retrieval (CBIR) traditionally focusses on the development of robust and efficient feature extraction, pattern recognition, and indexing techniques. A direct motivation for applying automated feature-based methods is the reduction of the effort of manually annotating image data using keywords and the complexity of manually categorizing images. However, the performance of traditional CBIR systems is mainly impaired by the mismatch between *low-level* features and their *high-level* semantics. The reason for this gap lies in the fact, that similarity between images is typically determined by applying a distance metric on a feature space, where only low-level features like color, texture, or shape are considered. Although these features can be used for similarity computation between images, they cannot adequately reproduce the human visual perception and interpretation ability. Hence, the linkage of low-level features to high-level concepts is solely possible in restricted application domains, like eye detection or finger print recognition.

Due to the importance of semantic meaning in image retrieval, the semantic modeling of image/multimedia contents facilitating domain specific reasoning has become indispensable in advanced CBIR systems. For that purpose, the manual assignment of annotations for the description of the data is performed more and more in both professional and personal retrieval applications. Existing annotations classify the data into semantic classes and can be used to facilitate textual or conceptual queries in large image repositories. Although manual approaches take away the uncertainty of fully automatic annotation, they require a high effort in exchange. As a consequence, methods for the semi-automatic annotation which combine the analysis of visual features and the manually performed description of image data are required. Furthermore, methods for supporting a consistent content annotation and the management of annotation data, including the unification of subjective annotations created by users having different background knowledge are of great demand.

The aspect of *'semantics'* in content-based retrieval plays an important role in several domains, like medical applications, television technologies, museums guidance, publishing companies, or military purposes. Some examples are:

Medical Applications: The identification and extraction of biomedical objects for their counting and conducting miscellaneous measurements is one of the main tasks in medical applications. Here, the semantic gap is represented by an object description delivered by a biomedical expert in natural language and its logical feature representation. The most frequently applied approaches in the medical area are based on building a classifier from training examples which will assign the image regions to one of the predefined classes. For example, in case of a binary classifier (e.g. SVM [CST00]), the extracted feature vectors are classified into *normal cell* or *abnormal cell* category.

TV and New Media: Institutions, like publishing companies, news channels, or advertising agencies are reliant on search possibilities at highest abstraction level, like thematic and concept based retrieval techniques. An efficient semantic classification and annotation is essential for the management of the available data.

Internet Applications: Due to the continuous increase in internet usage, searching for images from the web analogously gains importance. Although there exists a wide variety of systems supporting textual queries, the majority of them treat textual information as a bag of non-coherent query terms without any semantic meaning. Hence, there exist several requirements for semantic based retrieval methods and automatic annotation techniques. Some examples of such web applications are platforms for video sharing such as YouTube.com [You] or photo sharing such as Flickr.com [Fli], which contain valuable user-generated metadata, describing web resources using people's own vocabulary. Hence, this *weak* annotation provides fundamentals for further research in order to transform this data into a well-structured, reusable and understandable knowledge base.

Geography & Meteorology: The application areas of geographic information systems, meteorology, or astrology also need novel algorithms for digital content extraction and efficient storage techniques for the resulting amount of geographic maps or satellite images. Another possible application is the annotation of archaeology or historic art archives. For example, *Lost Art Internet Database* [Los] is one of existing projects for the documentation of lost cultural property, which was set up by the Government and the States of the Federal Republic of Germany. It registers cultural objects which were relocated or seized from especially Jewish owners during the persecution under the Nazi dictatorship and facilitates institutions and individuals who have suffered such a loss to make an advanced or catalogue search from the database.

Generally speaking, the application fields cover domains involved with huge heterogeneous image/multimedia collections, whose content may be attached with semantic meaning to become understandable and interpretable both for the user and for the machine.

1.1 Contributions

This thesis deals with the outcome of the *semantic gap* existing in image and multimedia retrieval. Methods for efficient storage, management, and retrieval in multidimensional data at semantic level assist the accurate usage of these media data. For this purpose, the combination of issues from several research disciplines, like pattern recognition, information retrieval, and knowledge discovery are needed to fulfill the requirements of an effective IR system. Mainly situated in the gap between the physical low-level features and the semantic representation of data, this thesis comprises several relevant contributions for realizing semantic retrieval from image data repositories. To give the reader a synopsis of these contributions, they are summarized in the following:

- The first contribution of this work is a *framework for semi-automatic annotation* within an existing image retrieval system. The proposed framework includes besides components for the extraction of low-level features, methods for the incorporation of semantic knowledge into the retrieval process. The semantic information is represented by ontologies which are used for an interactive annotation of the image data. Furthermore, the annotation component serves as an interface for the user feedback, which is performed for the definition of concepts needed for semi-automatic annotation. The annotation component is tightly coupled with the retrieval component, which is responsible for the analysis of the logical structure of already annotated data. Since the projection of visual features into a finite set of semantic concepts presents a real challenge, a possible solution for this problem is presented and discussed.

- A further contribution is the handling of different users' perception of image contents. Thus, we propose a method for the unification and integration of different annotation schemes which is based on the transformation of the annotation data into a graph representation. This representation allows the visualization of the complex semantic annotation space with its concept relationships and correspondences between keywords used for the annotation. The discrepancy between the background knowledge of different users of the retrieval system, their subjectivity, and the varying target application domains encourage all the more the assignment of keywords at multiple abstraction levels. The resulting *information overload* complicates the semantic retrieval considerably. In addition, we show by examples how to integrate our method into probabilistic approaches for (semi-) automatic image annotation.

- The aspect of *multi context* in information retrieval systems presents a further contribution of this thesis. The previously presented contribution for keyword-based retrieval supported by conceptual knowledge (e.g. ontologies) provides nevertheless further unresolved problems, like existing differences in interpretation of image contents or inconsistencies in keyword assignments among different users. To solve this problem, a new definition of *contextual similarity* is introduced, which is used to automatically infer the context in which queries are posed leading to an attenuation of users' subjectivity in content description.

- Another approach for narrowing the semantic gap is *Relevance Feedback*. This technique presents a powerful and widely used method for improving content-based image retrieval allowing query reformulation (QR) considering the user's subjectivity and perception. As our contribution, we present a realization of a *Pseudo Query Reformulation* on top of a relational database. In our approach, the internal query reformulation which iteratively computes the relevance values responsible for the reordering of the query results, is performed solely by considering the *relative* distance between images. The particular aspect of our approach is the fact, that the involved functions, like result judgments, relevance computation and reordering of the results are implemented as *user-defined functions*, making the method highly suitable for web retrieval applications.

1.2 Outline of this Work

The outline of this thesis is as follows: **Chapter 2** presents the background required for this thesis with a general introduction to information retrieval, requirements for image and multimedia data and the present progress in narrowing the semantic gap in image retrieval.

In **Chapter 3** we present our framework for supporting semi-automatic annotation of multimedia data which is based on the extraction of elementary low-level features, user's relevance feedback, and the usage of ontology knowledge. This approach facilitates image annotation by computing the most likely and relevant content descriptors as a result of extracted low-level features and the comparison of annotations of similar images.

Chapter 4 represents a further aspect of our work, namely the encountered problems during the annotation process, like the existence of multiple levels of abstraction, incompleteness of annotation data, or differing users' subjectivity. To solve the latter problem, we introduce the *Multi-level Annotation Model* which considers the several levels of abstractions at which annotations can be assigned. Within the proposed Annotation Analysis Framework, a graph-based representation technique is used in order to transform the annotations into a form which is understandable for the machine by providing inference making facilities. Furthermore, this representation serves for the unification and integration of different annotation schemes. In addition, we demonstrate the incorporation of our representation method into the probabilistic image annotation.

Chapter 5 deals with the problem of existing differences in interpretation of image contents or inconsistencies in keyword assignments among different users. The problem is introduced as the *problem of multi-context*, which appears during annotation-based retrieval based on the usage of *one* global ontology in multiuser retrieval systems. To simulate this problem, multiple sources of information, which are modeled as different user profiles and annotation ontologies, are brought together in order to extract *contextual information*, and consequently to attenuate users' subjectivity occurring during querying and content describing. At the same time, the users' subjectivity serves as an instrument for semantic *query expansion* preventing the retrieval to fail in case of different perspectives on image collections. Hence, the user is facilitated to search through his own subjective view of semantic concepts, but concurrently additional query parameters are inferred from other existing models. To evaluate the context-based retrieval, a set of experiments on a real-world domain of sports images has been done. In a second evaluation we used news data which allows efficient derivation of 'annotations' and is thus proved to be suitable for validating the proposed query expansion method.

Chapter 6 presents a *Pseudo Query Reformulation* method, which improves the content-based image retrieval by query reformulation considering the user's subjectivity and perception. The feedback cycle is characterized by users' interaction with the system in which individual result tuples are evaluated as relevant or not relevant for a given query. In answer to this, the query parameters are modified to better reflect the information need. The subsequent experimental evaluation on an image collection demonstrates the effectiveness of the presented relevance feedback approach.

Finally, the thesis is concluded in **Chapter 7** with discussions of future research directions and other difficulties of narrowing the semantic gap in CBIR.

2

Background

2.1 Introduction to Information Retrieval

Information Retrieval (IR) deals with the development of models and algorithms for the representation, storage, and extraction of relevant information from unstructured data [SM83]. A first straightforward definition of an IR system was given in 1968 by Lancaster [Lan68]:

> *'An information retrieval system does not inform the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.'*

Information retrieval existed long before the development of the World Wide Web. The primary goals of IR techniques comprised indexing text and searching for useful documents from large document collections with unknown contents, e.g. [CH79, Sal68, CCH92]. The used data collections contained publications and library records, but soon involved other fields of research, like medicine, biology, and journalism.

In the beginning of the 1990s, since the Web has become a universal pot of human knowledge and has provided access to a huge amount of information for everyone, the algorithms developed in IR had to be adapted for the requirements of the Web search. By the new developments, the IR matured into systems that not only consider the cross linkages available on the Web but also incorporate the semantic knowledge representation of web pages (e.g. meta data), with the aim to allow the automatic inference and machine understanding.

Today, research in IR includes data modeling, document classification, the development of graphical user interfaces, and visualization. In response to further challenges of

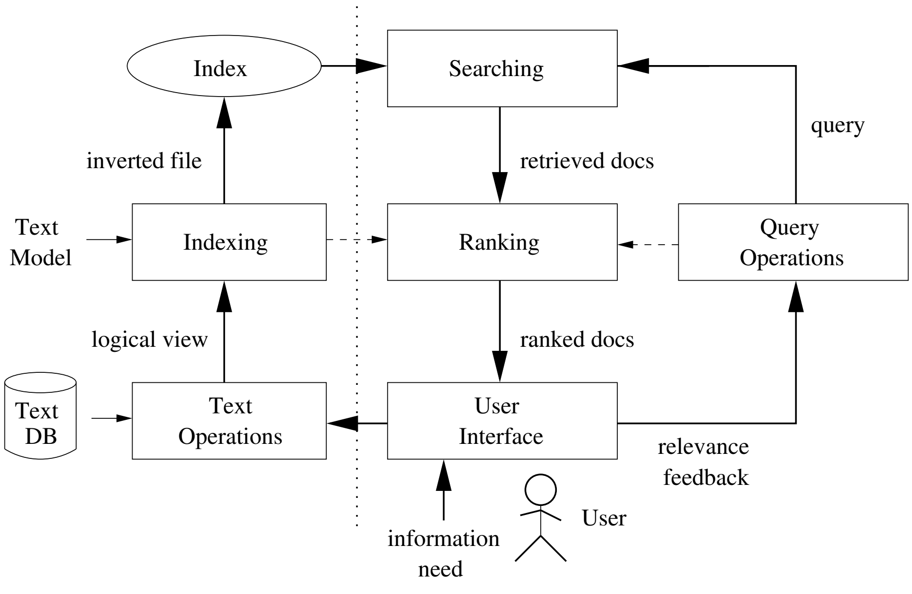


Figure 2.1: Typical Structure of an IR System (according to [BYRN99])

providing efficient information access, IR branched into related fields like personalized retrieval (e.g. using user profiles or search history), non-English language retrieval, data mining, and intelligent user interaction.

2.1.1 The Principles and the Structure of IR Systems

Independent of the application field, the information retrieval scenario can be coarsely characterized by the following steps [SM83]: First, the information need is formulated by the user through a query which is thus transformed into an internal representation understandable for the system. The system compares the query with all document representations in the data collection, ranks them according to their relevance and presents the result set to the user. In general, the goal of IR is to provide searchers data or media that will satisfy their *information need*. In order to describe the process of retrieval from textual data the following essentials must be first specified [BYRN99]: (a) document/text collection, (b) operations which transform documents into their logical view, and (c) the text model describing the structure of the documents.

For a detailed classification of the numerous methods involved the retrieval process, an overview of the structure of an IR system is demonstrated in Figure 2.1.

At the beginning of the retrieval process the user first specifies his or her *information need*. Already at this step, searchers are involved with the general 'vocabulary problem' as they are trying to translate their information need into a few search terms. The reason of this problem is the fact that several different words can represent the same concept (*synonymy*), and conversely, the same word can have multiple senses (*polysemy*). The search terms may consist of natural language statements or a list of keywords joined using Boolean operators *and*, *or*, and *not*. The query is entered into the system through the *user interface*, which serves as a visual component for the interaction with a retrieval system. Recently, some enhancements of this component have been developed for a better user assistance or to suit the evolving information need during the retrieval. For example, the interface may be extended by a thesaurus which will provide words related to the query and thus can be used for query expansion. Most bibliographic search systems, e.g. Citeseer [Cit], support navigational features such as browsing documents by specific content markers, like co-authors or citations.

By applying *query operations* which include parsing and transformation operations, the *query* is translated into the system's representation and can thus be understood by the retrieval system. Documents in the *text database* are frequently represented by a set of index terms or keywords. By using *stemming*, a transformation which brings words into their principal form, and by eliminating *stopwords*, the complexity of the document representation is reduced. These operations are called *text operations*.

When the text collection is large, methods must be applied to speed up the search. *Indexing* is a procedure of building suitable data structures over documents to make their access more efficient. The most commonly used indexing techniques are *inverted files* [BBH+87], *suffix arrays* [MM90, GBYS92], and *signature files* [LKP95].

As result to a query, the user obtains a set of documents, so called *retrieved documents*. The documents are displayed to the user, ordered according to their *likelihood* of relevance. The computation of relevance could be performed in different ways depending on the retrieval model, for example according to a distance function in the vector space model. In the optional *relevance feedback* loop, the user may examine the list of documents, and thus, the query can be reformulated by incorporating the user's judgments about the relevance/irrelevance of data in the initial result set.

Vector Space Model

The most popular representation of a document is the *vector space model* [SM83], alternatively known as the *bag-of-words* model. Its basic idea is to extract n content bearing unique terms from the union of all documents of the collection as *features* and then represent each document d as a vector $\vec{f}(d)$ of this feature space. Thus, the procedure of transforming a document into is vector space representation can be divided into two stages: First, non significant words, like function words (provided by a stop list), are removed from the *reference document vector*, so the documents will only be represented by content bearing words. The idea of removing *stopwords* is to leave out words that bear no content information, like articles ('the', 'a'), conjunctions ('and', 'or', 'but', 'since'), prepositions ('at', 'by', 'in'), etc. Normally, these words are characterized by a high frequency across the data collection and are thus not helpful for retrieval. Stopwords could depend on context, for instance, the word 'health' would probably be a stopword in a collection of medical journal articles, but not in a collection of articles from consumer reports. In the second step the document vectors (e.g. $\vec{f}(d_j) = [w_{1,j}, w_{2,j}, \ldots, w_{n,j}]^T$) are made up of *term weights*, each describing how characteristic a term is for a particular document. There are many alternatives for weighting terms, which are based on single scheme. There are three main factors: *term frequency*, *document frequency* and the *length normalization factor*. The principle of term weighting makes use of two criteria:

local. Terms that appear several times in a document are probably more meaningful than content words that appear just once and they are given a greater local weight. This local weight mirrors the importance of the term in a particular document.

global. The global criterion is based on the fact that words that occur in a handful of documents are likely to be more significant than words that are distributed widely across the data collection. It means simultaneously, that terms that appear in a large number of documents are not suitable for characterization of a single document.

As a general rule, local and global criteria are combined for weighting, resulting in the frequently used *tf-idf-weighting*. Let the text corpus consist of N documents, d_j be the j-th document, and let t_i be a term occuring in the data collection. Then the weighting $w_{i,j}$ of this term in the representation of document d_j is computed as follows:

$$w_{i,j} \;=\; \mathrm{tf}\,(t_i, j) \log\left(\frac{N}{\mathrm{df}\,(t_i)}\right), \tag{2.1}$$

where the term frequency $\mathrm{tf}\,(t_i, j)$ represents the number of times term t_i occurs in d_j and $\mathrm{df}\,(t_i)$ the document frequency of t_i, which denotes the number of documents t_i occurs in. The second factor is called the *inverse document frequency* (idf). Here, the logarithm is used to de-emphasize the effect of frequency. If a term occurs in a small number of documents, the inverse document frequency is high, and vice versa. Altogether, the value is a maximum when the term appears frequently in its own document, but rarely in other documents. The final step after weighting is called normalization. Long documents have usually a larger term set than short documents, which makes long documents more likely to be retrieved. To compensate this effect, document length normalization is often used. Thus, shorter documents are given more importance, and longer documents are imposed some penalty, so that every document has equal significance. *Cosine normalization* [SB88] is an effective technique. Every term weight in a document is divided by the Euclidean norm of the *tf-idf* weighted document vector. The three values - local weight, global weight, and normalization factor - determine the actual numerical value that appears in each non-zero position of the document vector. In this model, similarity between two text documents d_1 and d_2 or a query q and a document d_1 corresponds to the distance - or angle - between their vector representations. For example, the distance between two n-dimensional document vectors $\vec{d_1}$ and $\vec{d_2}$ is computed as follows:

$$\mathbf{d}(\vec{d_1}, \vec{d_2}) = \cos\alpha = \frac{\vec{d_1} \cdot \vec{d_2}}{\|\vec{d_1}\|_2 \cdot \|\vec{d_2}\|_2} = \frac{\sum_{i=1}^{n} w_{i,1} \cdot w_{i,2}}{\sqrt{\sum\limits_{i=1} w_{i,1}^2}\sqrt{\sum\limits_{i=1} w_{i,2}^2}} \tag{2.2}$$

where $w_{i,j}$ denotes the weight of the i-th term in document d_j.

By assigning non-binary weights to the terms, which are used for the computation of the degree of similarity, the vector space model supports partial matching and the resultant ranking of documents in the retrieval process.

2.1.2 Information Retrieval versus Data Retrieval

A related field of IR is the *Data Retrieval* (DR). According to it, the results of a query satisfy clearly defined conditions, formulated as regular expressions, formal logic, or by using query languages. As summarized in Table 2.1, the most important differences between DR and IR are the following: In DR, the transformation of the stored data into its meaning is done by using query languages. For example, records can be requested from a relational table, whose data is organized according to some well-defined syntax. Consequently, DR deals with modeling, organization, and the retrieval of data which is formatted in a way that makes it easy to manipulate and manage, for example when it is stored in databases, fixed-format files, or log files. Example 1.1 presents two

	Data Retrieval	*Information Retrieval*
Data	structured	unstructured
Matching	exact	partial
Relevance	binary	graduated
Error	sensitive	insensitive
Query	query languages	natural language
Inference	deductive	inductive

Table 2.1: Data Retrieval versus Information Retrieval (according to [Rij79])

queries which contrast IR with DR.

Example 1.1 *Querying.*
Data Retrieval: `SELECT * FROM books WHERE title LIKE ’Databases%’`
Information Retrieval: ’Search for all books which deal with principles of databases’

A great challenge for IR is the management and the automated information access to *unstructured or semi–structured data* not containing any ’meaning’ or *semantics*, but only providing implicit information which has to be interpreted by knowledge discovery algorithms or by the user first. The main characteristics of unstructured data is, that it does not possess any schema and therefore, it cannot be understood by looking at its meta data or its contents. For example, in the context of relational database systems, it refers to data that cannot be classified in rows and columns (e.g. images). Instead, images have to be stored as BLOB (binary large object), a universal data type available in most relational database management systems. Another unstructured data may include video, audio files, or web pages which might be semi–structured by offering the option of meta data.

Example 1.2 *Data Storage.*
Data Retrieval:

PubID	Title	Author	Year	Topic	Category
147	...	653	2005	Computer Science	Databases
...	...	...	...	...	...

Information Retrieval:

```
CREATE TABLE ‘gallery‘ (
  ‘id‘ INT(11) NOT NULL AUTO_INCREMENT,
  ‘title‘ VARCHAR(64) NOT NULL,
  ...
  ‘data‘ MEDIUMBLOB NOT NULL,
  PRIMARY KEY  (‘id‘)
);
```

In data retrieval, all returned documents which fulfill the query conditions (*exact match*) are returned as an unordered set. In IR, the similarity of a piece of information or a document d_1 to a given query q is expressed by *partial matching.*

$$IR : f(d_1, q) \rightarrow [0; 1] \tag{2.3}$$

Thus, the relevance to a particular query is assigned both by the presence or absence in the result list, and the relevance degree by the ranking in the ordered list. Since the information content of the result objects is only implicitly derivable, relevance represents as well a subjective judgment, which mirrors whether the results fulfill the initial query aims and/or satisfy the users' information need. Furthermore, extractable aspects like matching topic, reliability of the source, or up-to-dateness might be considered in the relevance computation.

Another distinction can be made in terms of error sensitivity. The similarity estimation considering several characteristics of the data, results in an error tolerance for the query formulation process. The query language for DR is generally artificial with formally restricted syntax and semantics. For example, relational databases allocate the *Structured Query Language* (SQL) for the formulation of queries. IR rather uses the natural language (e.g. query-by-example or terms) to formulate the information need.

Furthermore, a data retrieval query provides a result set which presents a complete specification of the information need, in IR the result set is invariably incomplete. The reason for this distinction between the two paradigms is the process of decision making. Inference is the ability of deriving new conclusions from existing facts, resulting in new knowledge. The inference used in data retrieval is of *deductive* kind, that is, the extraction of particular facts (or conclusions) from the general is done by selecting tuples from relational data. Here, the conclusions inferred from a valid deductive inference are always true if the premises are true. In the IR, conclusions are acquired by the *inductive* reasoning, which means that specific observations are inferred to generalized conclusions. The conclusions may be correct or incorrect, since the premises are specified with a degree uncertainty. For example, in a query-by-example process, the query object serves as an example which expresses the user's information need. Now, the inductive reasoning manifests itself (as illustrated in Figure 2.2) by the inference from the characteristics of this one relevant example to the properties of all relevant documents.

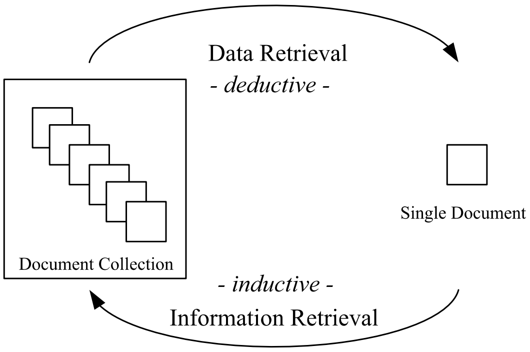


Figure 2.2: Reasoning in Data and Information Retrieval

2.2 Content-based Retrieval of Multimedia Data

Content-based Image Retrieval (CBIR) has emerged as a special research area of *Multimedia IR*, with the thrust from various disciplines, like databases, computer vision, artificial intelligence, and image/signal processing. Hence, the image retrieval process poses several interesting challenges for each of the research fields. In general, multimedia IR differs from traditional IR in many aspects suggesting new necessary techniques which exceed the methods used in traditional IR systems. First, the complexity of multimedia data (e.g. web pages, image data, video sequences, and audio files) requires the extension of database management systems by functionalities for representing, storing, and processing multimedia objects. Also the various application fields of multimedia form a huge diversity of systems with different requirements and underlying algorithms. In the first instance, such systems are targeted to support the user in the query formulation process, to provide techniques for content extraction, and for an efficient similarity computation between the representation of the multimedia data and the posed query.

Table 2.2 gives an overview of properties and differences between Text IR and CBIR. Beside the complex data representation and the resulting need for huge storage capacity, techniques for extracting and selecting features from a variety of different data types have to be implemented in CBIR.

- **Querying.** The querying is based on a similarity approach, and is mostly processed as *query-by-example* [ZZ00], respecting a reference image to be provided by the user for the initialization of the search. There exist many different possibilities to specify a query, like attribute-based (for example, conditions on the attribute *'color'* of an image), content-based query (*'find all images containing a car'*), and query by structural elements (*'find all multimedia objects containing audio files'*).

- **Features.** Different from traditional text-based retrieval which uses a set of terms for the description of documents, CBIR classifies and searches images

Aspects	*Text Retrieval*	*Multimedia/Image Retrieval*
Storage	negligible	huge
Data Representation	terms	complex representation
Features	weighted term occurrences	general and domain specific
Querying	query-by-terms	query-by-example
Subjectivity	not given	crucial

Table 2.2: Differences between Text and Multimedia Retrieval

according to similarities of automatically extracted visual features, such as color, texture, shape, and structure. These automatically extracted features are the basis for both, basic similarity computations between images, but also for other heuristic retrieval methods, like [ISF98], or machine learning approaches with relevance feedback [HROM98, RHM97, RHM98].

- **Subjectivity.** Although in current CBIR systems, low-level features are widely used for the similarity computation between image or audio data, they cannot adequately reproduce the human visual perception and interpretation ability. By applying a distance metric based on features extracted from the data, there appears another important issue, namely the *retrieval subjectivity*, which represents a big drawback in CBIR systems. This subjectivity results from the richness of human interpretations in several retrieval steps, like querying, indexing, users' keyword assignments, or appraisal of the retrieval results. In summary, the performance of traditional CBIR systems is impaired by the mismatch between low-level features and their high-level semantics, a phenomenon which is known as the *semantic gap* [ZG02].

Several further aspects have to be considered in image retrieval: For example, data modeling, multidimensional indexing, and efficient querying in high dimensional data play an enormous role. For efficient finding of data objects which are similar according their contents, and thus improving the query performance, multidimensional index structures, like the *X-tree* [BKK96] or the *R^+-tree* [SRF87], have to be used. These types of indexes divide the original data space into sub-regions according to the distribution of the data objects inserted into the tree. A detailed survey of several indexing methods is given in [BBK01]. Nevertheless, these methods are proved to work well for low dimensional problems but they degrade drastically as the dimensionality increases. As solution for overcoming this curse of dimensionality, the idea is to compress the data into a few dimensions [CM00, KAAS99] by applying data transformation methods such as *principal component analysis* [Jol86] (PCA). The dimension reduction approaches rely on the fact, that, in many cases, not all the present dimensions are important for understanding and modeling the meaning of the data.

Feature Representation and its Usage for CBIR

Features extracted from image data are classified as *general* features and *domain specific* features [RHM99]. General features describe standard properties of the data, like color distribution, texture, and shapes. Application-dependent features are implemented for a certain domain-specific purpose, like face/eye recognition or motion recognition. Since the perception of features is subjective, their characteristics can be described from several perspectives, that means for any given feature there exist multiple representations which require new adapted similarity measures.

The first CBIR systems have been based on the extraction of features like color, texture, shapes (regions and contours), spatial layout, object motion, etc., which are summarized in [GR95]. In the easiest case, shapes are determined by edge detection algorithms [MCR02, Sap06] or by region-based grouping [LS01]. Another widely used approach are the *active contour models* [KWT88] (snakes), which are based on an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. A short summary of commonly used features and their possible representations is given in Table 2.3.

Type	Representation Examples
Color-based Features	Color Histograms, Correlograms, Color Moments, Color Coherence Vector
Texture	Coarseness, Contrast, Directionality, Regularity Fourier Power Spectra, Markov Random Fields
Region-based	Salient Regions, Region Moments, Spectral Descriptors
Contour-based	Edge Flow, Representative Points, B-Spline, Shape Signatures
Layout-based	Spatial Relations, Axis Orientation

Figure 2.3: Feature Types and their Representations

In most cases, CBIR approaches are based on a combination of various features which are weighted appropriately, and the basis for the computation of complex features is given by a composition of representations. For example, in order to determine salient regions in an image, homogeneous regions based on color, texture or moments have to be characterized. Also texture-based methods can be effectively improved by a combination of both edge information and gray level co-occurrence matrix properties (e.g. in [ZH06]). There are thousands of enhancements of the feature extraction methods which go beyond standard pixel-based image processing. For example, the statistical *expectation maximization* (EM) can be used for the segmentation [CBGM02, CTB$^+$99] of image contents into a set of uniform regions that are coherent in color and texture. The features are extracted for each of the computed regions (so called 'blobs'), which roughly correspond to objects. The advantage of this method is that queries are performed at the level of objects rather than global image properties.

Further advanced methods described in the literature are machine learning approaches which have been not only applied for retrieval purposes but also to narrow the gap between the low-level features and the high level semantic. For example, *support vector machines* [TC01] (SVM) have been successfully used in the relevance feedback process, in which the user returns the system a set of relevant/irrelevant data examples. These examples serve as feedback for the SVM-algorithm which learns a boundary to separate the irrelevant data from the images which satisfy the user's query.

In the following paragraph, a selection of two features and their representations is presented:

Color Feature

The color feature is a primitive, but one of the most frequently used feature in many image retrieval systems because of its robustness to background noise and orientation invariance. In the *RGB color space* a color is represented by a combination of the levels red (r), green (g), and blue (b).

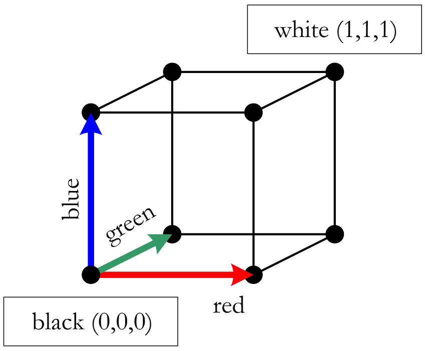


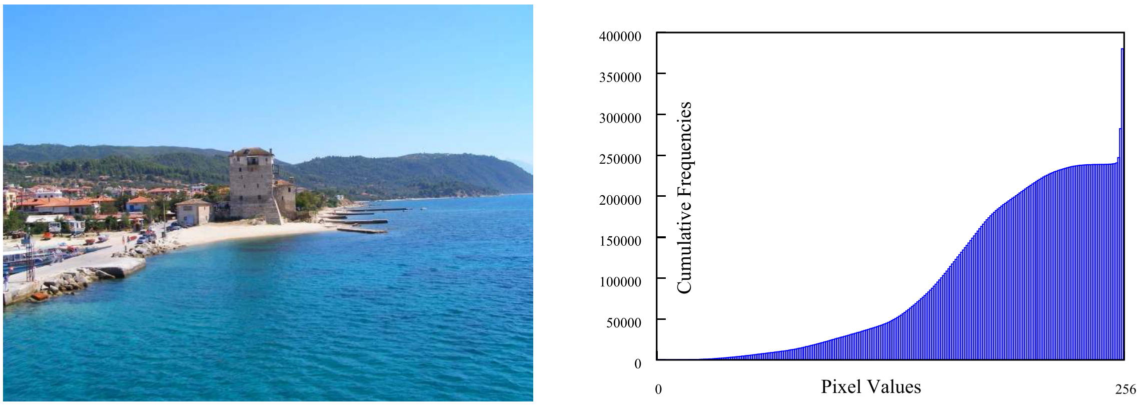
Figure 2.4: RGB Color Space

As illustrated in Figure 2.4, each color is defined by a point within a 3-dimensional cube and is characterized by the triplet (f_r, f_g, f_b). Normally, a component of such a triplet ranges from 0 to 255, but can arbitrary be normalized to other ranges (e.g. [0,1]).

A more compact representation is provided by dividing the RGB cube into a smaller set of bins $b_1 \ldots b_n$ (e.g. $4^3 = 64$ bins) in order to reduce the dimensionality of the feature vector. Now, a histogram can be constructed by determining the number of pixel contained in each of the bins. Additionally, the influence of the image size might be eliminated by dividing each value by the number of pixels in the image. Another modification is to consider the *cumulative histogram* representation, which represents the probability to find a pixel that has up to a certain intensity ν. The form of the curve gives an indication how uniformly the color levels are distributed and this information can afterwards be used as input for image equalization methods [PAA+87]. In the cumulative histogram, the value of $H(i)$ at each of the L grayscale levels is computed using the following equation:

$$H(i) = \frac{1}{N \times M} \sum_{\nu=0}^{i} h(\nu) \qquad (i = 0, \ldots, L-1) \qquad (2.4)$$

where $h(\nu)$ presents the number of pixels with intensity ν and $N \times M$ are the image's dimensions. In order to apply this method to color images, the values of the three RGB components might be piled in separated bars. Figure 2.5 represents the cumulative histogram for the blue color band.



(a) Color image of 712 x 534 pixels

(b) Corresponding cumulative histogram

Figure 2.5: An Example of the Cumulative Histogram of a Color Image

Depending on chosen histogram representation, a method for the similarity computation between two images have to be selected. In the simplest way, the distance between images I_q and I_p is computed by the Euclidean metric:

$$dist_{hist}(I_q, I_p) = \sum_{j=1}^{n} |h_q(b_j) - h_p(b_j)|^2 \ . \tag{2.5}$$

In this equation, n denotes the number of bins, and $h(b_j)$ represents the histogram value for a given bin b_j. Another similarity measure for the color histogram is the L_1 metric [SB91], where two histograms are intersected to find color coverings in the color values. As extension to this approach, the L_2-metric has been introduced in [Iok89], which considers additionally the similarities between close but not identical colors.

Although color histograms are easy to compute, on the other hand they bring problems for image indexing and retrieval. First, they require quite large memory, since color histograms consist of from 64 to 256 bins. This large histogram size makes it rather difficult to create an effective database indexing scheme. Second, they do not include any spatial information; hence they are prone to false positives. Third, they are sensitive to small brightness changes, and therefore are liable to false negatives as well. Finally, they are basically incompetent to support partial matching of image contents. Partial matching is essential to many image retrieval requests, for example the query such as *'find images that contain a green lawn while ignoring the rest part of the images'* could not be executed without partial image matching abilities.

Beside the histogram-based representations there are other color-based approaches presented in the literature. Some popular characterizations of color are *color moments* [SO95] and *color correlograms* [HKM+97]. The former representation is based on the assumption that the distribution of color in an image can be interpreted as a probability distribution for each color channel. Thus, the moments of a such distribution (e.g. *mean*, *variance* and *skewness*) can be used as features. Color correlograms are robust to large appearance changes, like modification of viewing position or camera zoom, by describing the global distribution of local spatial correlation of colors.

Texture Feature

Another feature which can be easily perceived by humans is the *texture* in an image. In general, texture is a property which expresses the (in)variance of certain statistical features that are periodically or quasi-periodically distributed over a region. There are two widely used approaches to describe the texture of a region, namely *statistical* and *structural* methods. The statistical approach considers that the intensities are generated by a two-dimensional random field. These methods are based on spatial frequencies and yield characterizations of textures, for example as smooth, coarse, or grainy. Examples of statistical approaches are texture statistics such as *moments* of the gray level histogram, *gray level co-occurrence matrix*, or *fourier texture analysis.* In structural approaches, texture primitives are extracted as the basic elements of a texture and are used to form more complex texture patterns by applying production rules, which specify how to generate texture patterns.

The *gray level co-occurrence matrix* (GLCM) is often found a fairly good texture analysis method which uses a set of features, like *energy*, *entropy*, and *maximum probability*, as texture description. The aim of the GLCM is the characterization of gray level variances in the neighborhood of a certain pixel. Thus, the preliminary thoughts are that texture can be adequately described by gray level distribution of pixel pairs having the distance d and the angle α. Figure 2.6 gives a schematic construction of a GLCM(α, d) as an $L \times L$ matrix ($L = 256$). The values $x_{i,j}$ stand for the occurrence frequency of pixel pairs having gray level values i and j. That means, each element $x_{i,j}$ in the resultant GLCM presents the number of times a pixel with gray level i occurred in the specified spatial relationship (defined by α and d) to a pixel with level j in the input image. Hence, the following features can be inferred from the matrix:

$$\mathrm{GLCM}(\alpha, d) = \begin{pmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,L-1} \\ x_{1,0} & x_{1,1} & \cdots & \cdots \\ \vdots & \vdots & \ddots & x_{L-1,L-1} \end{pmatrix} \tag{2.6}$$

Figure 2.6: Gray Level Occurrence Matrix (GLCM) in Respect to α and d

Energy. The property of *energy* is a measure for regularity.

$$\mathit{energy} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\left(\frac{x_{i,j}}{R}\right), \text{ where } R = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} x_{i,j}. \tag{2.7}$$

Regular patterns have a certain number of highly repeated pairs of gray levels. This results in a GLCM containing a few high values and many small values, and thus Equation 2.7 will determine a high energy value.

Contrast. When the contrast of a texture is high, it is obvious that the gray level difference between two pixels is high. In the following Equation 2.8, the values of the GLCM are multiplied with $|i-j|^2$, resulting in a higher weighting of values far away from the matrix diagonal.

$$\mathit{contrast} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} |i-j|^2 \left(\frac{x_{i,j}}{R}\right) \tag{2.8}$$

Homogeneity. This measure is the contrary part of contrast, which manifests itself by higher weighting of the diagonal values, and an attenuation of values which are far away from the matrix diagonal.

$$\mathit{homogeneity} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \frac{1}{1+|i-j|}\left(\frac{x_{i,j}}{R}\right) \tag{2.9}$$

In general, texture features are very difficult to detect and characterize, since periodical repetition underlies significantly random oscillations. Furthermore, textures are of hierarchical structure, which means, that their appearance may change under varying amplification factors (e.g. patterns like *curtain* or *wood*), complicating the categorization of a certain pattern to classes. Considering the computational aspects, the matrices may become very large, particularly if images are composed of a large number of gray levels. Additionally, their static nature makes them inefficient since a re-computation is needed when the image's dynamic range varies. A further problem is the amount of *zero information.* Since texture is usually measured in a small region, a large number of entries are zero contributing nothing to the texture description of the region. Thus, the computational time of the texture feature extraction operations includes also the time for processing those entries.

The presented features contrast and homogeneity are rather basic representations, which have been overcome by other sophisticated methods for texture analysis. The recent literature has shown that approaches which are based on *markov random fields* [PH92] have a great potential for texture analysis. In [KBC04, PL03] *wavelet transformation* is used for computing the energy and standard deviation of twelve different directions in an image. The evaluation of this approach shows that the retrieval performance is superior relative to the performance obtained using the other existing retrieval methods. In addition, approaches using *Gabor functions* and *filters* are regarded as one of the excellent methods for texture segmentations. To overcome the problem of rotation invariance of Gabor filters, a method which extracts local texture features from blocks of an image and applies a rotationally *invariant Gabor wavelet filter* is presented in [SKKP03].

2.3 Semantic Gap in Retrieval Applications

Although numerous advances have been made in the design and performance improvement of retrieval systems, there are still open research issues to be solved. The most important aim is to overcome the *semantic gap* [ZG02], which can be seen as the discrepancy between human perception and the features determined by automatic extraction algorithms. In [SWS+00] the semantic gap is described as:

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

Particularly in image search, the performance of traditional CBIR systems, like [SC96, PPS99, NBE+94], is considerably impaired by the mismatch between *low-level* features and their *high-level* semantics. The most critical aspect of conceptual queries at semantic level is the fact, that similarity between images is determined by applying a distance metric on a feature space, where only low-level features like color, texture, or shape are considered. Furthermore, a reliable linkage of low-level features to high-level concepts is exclusively possible in restricted application domains, like eye detection or finger print recognition. Based on these facts, approaches for narrowing the semantic gap in CBIR can be coarsely divided into three classes, solving the problem from different angles:

Bottom-Up. The bottom-up approach tries to automatically build bridges between low-level features and the semantic classification of the data. Some representative methods are image clustering and automatic categorization, or the automatic image annotation, which is predominantly based on the correlation computation between visual features and semantic labels.

Top-Down. Another promising method to bridge the gap is to restrict the application domain of the image collection and to specify a finite content description vocabulary, which is assigned to the data in form of annotations. Furthermore, knowledge about data instances and their relations are modeled using ontologies in order to facilitate query expansion and inferences at semantic level.

Hybrid. The combination of the bottom-up and top-down provides methods which incorporate users' knowledge and perception into the retrieval process (e.g. by users' interaction) in order to adapt the internal representation of the data. Examples of these approaches include personalization aspects or relevance feedback, which will be presented in Section 2.3.1.

2.3.1 Personalized and Adaptive Retrieval

Static retrieval methods suffer from the incapability to satisfy heterogeneous needs of many users. One possible solution to overcome this negative effect is to develop systems (e.g. [SS98a, PPPS03, CSB$^+$03]) with the ability to adapt their behavior to the goals, tasks, interests, and other features of individual users or groups of users [BM02]. The so called *adaptive* IR has become increasingly important and describes the process in which the search is adapted to users' needs/preferences with the objective of optimizing the search. In this context, *personalization* is considered as a subset of adaptive IR [Bru96] and means that the system knows users' preferences (profiles) and changes its behavior accordingly. For example in the case of an IR system which knows that the user is not interested in vehicles, it will not return results dealing with cars on a query with the term 'jaguar'. In summary, personalization of IR is explicitly concerned with user-based factors, like level of knowledge, interests, or available hardware parameters (see Figure 2.7).

Adaptivity is the property of *automatic personalization* and means that the system creates a model of the user using heuristic and probabilistic approaches. The adaptation is based on non-user factors, like users' interaction or implicit feedback. An adaptive system uses the interaction to acquire knowledge about a user and estimates changes in his information need over time. As a consequence, the solely source of information is composed by users' events and actions which are exploited to build a *user model.* This model is subsequently used to modify search queries and to make new search decisions such as re-searching the document collection or restructuring already retrieved documents.

Adaptation can manifest itself in different ways. A rough distinction can be made between *search-based*, *browsing-based* and *presentation-based* adaptation. The search-based adaptation is conducted in the background during user's search for relevant information. The system then analyzes both the search parameters, like search terms, and the user's model to identify the most relevant documents satisfying the user's current information need.

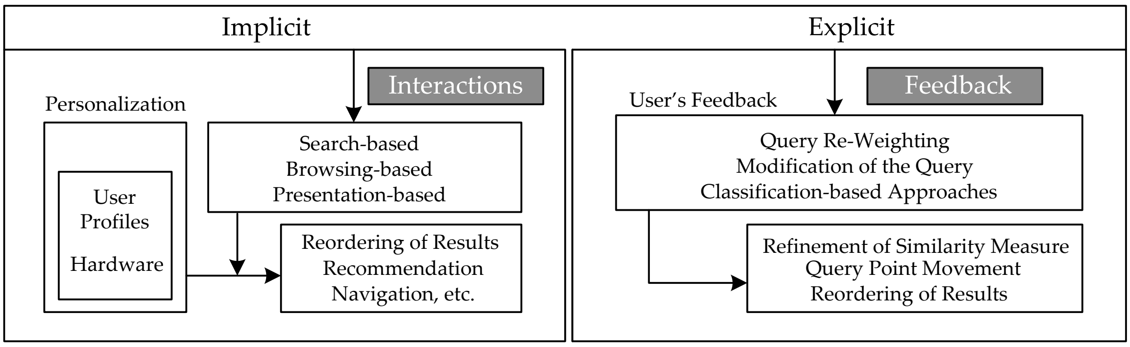


Figure 2.7: Implicit and Explicit Approaches in Adaptive Systems

The browsing-based adaptation, introduced by [Bru96], deals with the browsing-based access to web information. Here, the user is supported in browsing, that means, during the navigation from one document to another, the system manipulates the links, e.g. by hiding, sorting, or highlighting relevant documents, to guide the user adaptively to most relevant web pages. Some advanced extensions of this approach comprise automatic document classification [AT97], visualization of relevant links, or link recommendation [SS98b] by dynamically learning the user's areas of interest.

The third technology has some deep roots in the research on *adaptive presentation* in intelligent systems [Par88]. In opposite to classical IR paradigm, this approach does not localize relevant information, but organizes contents and presents them according to users' profiles. One representative of this approach is PowerBookmarks [LVC$^+$99], a system for personalizable organization, sharing, and managing of web resources. In order to index and classify web resources, it parses metadata from their URLs, and supports advanced query, classification, and navigation functionalities on collections of bookmarks.

In contrast to implicit feedback which constructs inferences on what is relevant from interaction, *relevance feedback* [SB90, OBM03, PMO99, RHM98, HROM98, RHM97] (RF) approach is based on the *explicit* evaluation of retrieval results. The aim of this approach is to refine query results by taking users' expertise into account and to adjust the query towards the existing information need. At the same time, RF should attempt to minimize the amount of interaction between the user and the IR system required to achieve satisfying search results. Although several image retrieval systems including effective feature extraction algorithms [DR01, SNL02, FSN$^+$95] have been proposed in recent years, none of them can capture the hidden high-level semantics successfully. To address this issue, RF is used as a powerful tool to narrow the semantic gap between low-level features and high-level concepts. In systems supporting this technique, e.g. MARS [ORC$^+$98], the relevance feedback cycle (Fig. 2.8) is initialized by users' selection of a set of images that appears to be relevant to the initial query.

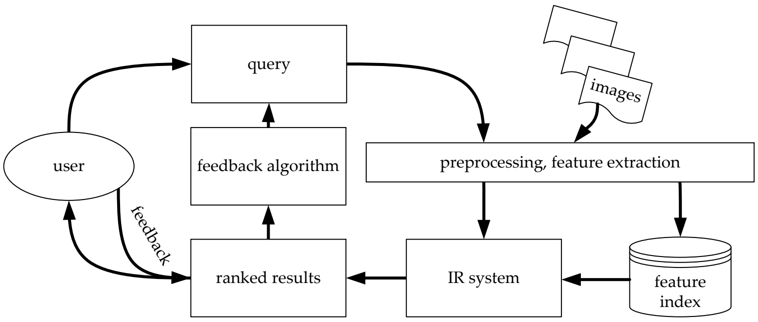


Figure 2.8: Relevance Feedback Cycle in CBIR

The subjective user evaluation (by marking images as relevant or irrelevant) serves as input for the feedback algorithm which uses the features derived from the selected tuples to revise the original query in the next search iteration, subsequently leading to an improvement of the retrieval results. This cycle of relevance feedback is repeated until the user is satisfied with the results.

The most commonly used feedback algorithm which modifies the query is *query re-weighting* with its basic idea to learn *feature weights* from relevant images (or/and irrelevant images) and use them as new parameters for the subsequent query [BS95, Sha95]. In the broader sense, it represents an attempt to map interesting high-level concepts to system's low level features. In [HROM98] an object model has been presented which supports multiple representations of the image contents and query objects (see Figure 2.9). According to this approach, weights exist at three levels, namely W_i, W_{ij} and W_{ijk}, which are associated with features f_i, their representations r_{ij} and the components r_{ijk} respectively. For example, a feature f_1 may stand for the feature *color*, $r_{1,5}$ for a possible representation, e.g. its *histogram.* Since the representation itself may be a K-dimensional vector, i.e. $r_{ij} = [r_{ij} \dots r_{ijk} \dots r_{ijK}]$, its components can be weighted individually.

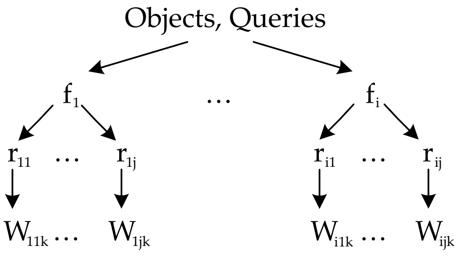


Figure 2.9: Multiple Image and Query Representation in a Re-Weighting Approach

At the beginning of the relevance feedback cycle, all weights W_i, W_{ij} and W_{ijk} are initialized giving every entity the same importance. The user's information need, which is represented by the query Q is distributed among the features f_i and their representations r_{ij}. The objects' similarity S^Q to the query Q in terms of r_{ij} is calculated according to the corresponding similarity measure m_{ij} and the weights W_{ijk} [HROM98]:

$$S^Q(r_{ij}) = m_{ij}^Q(r_{ij}, W_{ijk}) \tag{2.10}$$

The received representation's similarity value is propagated through feature level, resulting in a feature's similarity value $S(f_i) = \sum_j W_{ij} S(r_{ij})$. Afterwards, the overall similarity $S = \sum_i W_i S(f_i)$ is obtained by summing the weighted $S(f_i)$ values. After the objects in the database have been returned according to their similarity, the user assigns to some or each of the retrieved results a score of relevance (for example values between $[3, -3]$). As last step, the system updates the weights (a detailed description can be found in [HROM98]) of the query according to the user's feedback, and perform a new iteration with the adjusted Q.

A further relevance feedback approach allows users to modify the query point and thus refine the query representation. An established method for refining the query is given by *query point movement* [RHM97], which assumes that there exists an ideal query point which has to be estimated by the users' feedbacks. Several *classification-based* RF approaches have been proposed in the recent years in order to conduct effective relevance feedback for image retrieval. For example, [TC01] proposed the use of an *active learning* algorithm based on a support vector machine which quickly learns a boundary that separates the images that satisfy the user's query from the rest of the data collection. As an extension of this two-class (relevant and irrelevant) learning problem, a multi-class form of relevance feedback has been proposed in [Pen03]. Here, for a given query, the local relevance of each feature dimension is determined based on Chi-squared analysis using information provided by the multi-class relevance feedback. This information is then used to flexibly customize the retrieval metric.

In order to establish a relationship between annotation-based multimedia systems and the presented approaches, adaptivity and personalization can be applied in the following manner:

- In the approach for semi-automatic annotation, keywords for the description of image data can be automatically extracted by considering results of the feedback cycle performed by the user. For example, the initial search keywords may be automatically added to images that received positive feedback, which will facilitate keyword-based retrieval in the next iteration.

- Annotations created by the user categorize image data into semantic concepts.

- In retrieval systems supporting semantic search, adaptation is done at multiple levels. In the first instance, the visual characteristics of an image class could be updated by analyzing similar images belonging to this class. By explicit relevance feedback images with similar semantics could be automatically grouped into the same semantic class.

2.3.2 Describing Semantic Content by Annotations

To this day, the gap between low-level features and high-level concepts still presents an unsolved problem in CBIR approaches. Several techniques have been proposed in past years, e.g. in [NBE^{+}94], most of them are based on the query-by-example approach, which provides as query result a set of images due their similarity to a user provided image object. More sophisticated approaches use relevance feedback from the perspective of machine learning where the system's performance is enhanced by user's interaction and query refinement. However, there are still many unresolved issues in content-based systems. The first disadvantage is the fact, that these approaches require the user to query based on low-level features like color, texture, and shape which they are not familiar with. These methods do not take into account that an advanced and fully functional retrieval system would require support for queries at semantic level. Furthermore, CBIR retrieval methods are mostly restricted to particular application fields (e.g. medicine, geographic information systems) causing the assignment to heterogeneous image collections to fail in terms of accuracy.

Due to the importance of the *semantic meaning* in the retrieval process the annotation of image data becomes indispensable in both professional and personal image retrieval applications. In summary, the motivation for assigning annotations to image data includes the following aspects:

- Users are highly interested in querying images at the conceptual and semantic level, not only in terms of features like color, texture, or shape [TPCR04].
- An extensive annotation of the data facilitates keyword-based search in large image repositories.
- In general, users would like to pose semantic queries using keywords or concepts and find images relevant to those semantic queries. For example, this would make queries like *'find me all images of sunsets at sea'* possible.

A fundamental requirement for semantic annotation is to provide dynamic data structures, formalized as a *semantic data model*, which is used for the conceptual description of media contents [GMY95]. In the easiest case, contents are annotated using a set of independent keywords itemizing objects found in the image. Since 2001, motivated by *Semantic Web* [BLHL01b] technologies, the research in ontology-based image annotation has boosted. Here, the annotation domain is characterized by a conceptualization of knowledge in terms of entities, attributes, and relationships [SBWR06].

Several studies, such as [SDWW01, HSWW03] demonstrate the usage of specialized ontologies in art and private photo collections in order to perform domain-specific annotations.

A further factor to consider is the *pragmatics* of an image, which is defined by its relationship to the interpreter and depends on his point-of-view. Pragmatics considers the specific usage of the data. That means that some additional knowledge which has to be provided by the user is needed for the semantic categorization of the image. For example in news agencies, the image repositories are frequently scanned for a suitable image as an illustrative supplement to the authored news text. In this case, only the topic of the query is defined, not the image contents. The search at pragmatical level, which is really demanded in many application fields, is not sufficiently supported in non-textual retrieval.

The core element of an image retrieval system is the underlying knowledge representation model. In the literature, several image data models and description schemes have been proposed which consider a certain number of representation levels. One of these approaches is the *VIMSYS* [GWJ91] data model, which represents images as 4-layered objects or the *EMIR2* [Mec95] which gives description for an extended image data model used for retrieval purposes. For the description there exist various *image data models* and *description schemes*, e.g. [GS00, SOCP99], which allow to define relations between entities and to capture the knowledge of particular application fields. The most important requirement for the data model is its expressiveness to qualify the structure and contents of the underlying image data, data objects, and relations among them. The design of an appropriate image data model will ensure smooth navigation among the images in a database system and a fast access to all the logical representation of image data.

Manual, semi-automatic, and automatic Annotation

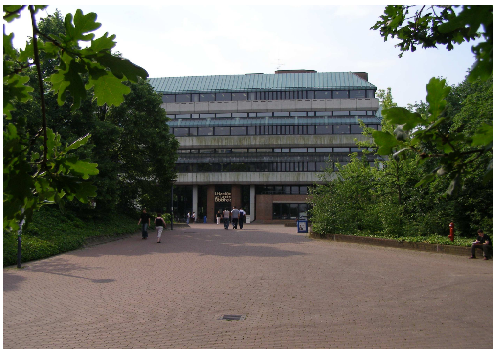
The semantic gap is not the unique explanation for the difficulties encountered in retrieval by content, but is broaden by the use of incomplete or confusing descriptions of multimedia contents. To reduce this problem, users should attempt to assign descriptors that are both rich and faithful. In the following, we will characterize the manual, automatic, and semi-automatic annotation with their assets and disadvantages.

Problem	Possible Solution
incomplete/confusing descriptors	linguistic processing
time consuming	semi-automatic annotation
users' perception differs	controlled vocabulary

Table 2.3: Problems in Manual Image Annotations

Manual Annotation

Manual annotation [GZCS94, HSWW03] is the process of assigning descriptive keywords to images from a controlled or uncontrolled vocabulary, which is accomplished by users themselves in order to manage their personal multimedia contents and to facilitate public search. In recent years, there has been a rise of manual image annotation systems accessible for many online communities, like PhotoStuff [HWGS+05] or Flickr [Fli]. In the latter system, for each uploaded image the user is encouraged to create a free-text annotation, which forms a central component for the retrieval and discovery of the shared contents. This approach uses the benefit that traditional text retrieval techniques can be applied for image data. Figure 2.10 shows a sample image described by a set of keywords reflecting the image's semantic concepts. Although manual annotation takes away the uncertainty of fully automatic annotation, it requires a high effort in exchange and keywords do not always capture the content of images appropriately. Another weak point is that indexers often use different descriptors and their perception can be influenced by person's mood, knowledge, or other factors, providing a varying annotation quality. As result of the varying user's subjectivity, similar images have often few keywords in common, instead of having a large number of overlapping keywords. Another problem is the so called *vocabulary problem*, which means that a user will probably assign different words to the same concepts during a certain time period. As a consequence, this vocabulary disagreement leads to inconsistencies in the keyword assignments resulting in ineffective retrieval. As a summary, the problems encountered in manual annotation are presented in Table 2.3.



keywords: building, entrance, people

Figure 2.10: Sample Image and its Annotation

Automatic Annotation

The most popular approaches for automatic image annotation, e.g. [CCS03, JLM03, DBdFF02], are based on feature extraction and the correlation computation between the visual features and the used annotation vocabulary. Another probabilistic approaches associate words with image regions by using a *co-occurrence model* (e.g. in [MTO99]) analyzing the co-occurrence of words with image regions created using a regular grid. Some sophisticated approaches use *stochastic models*, like [GIK05] or *learning methods*, like [CV05, KJC04] where the annotation is reduced into a supervised/unsupervised learning problem. In particular [DBdFF02] proposed to use the *transition model* [BDPDPM93] to learn the mapping between region types and keywords, which are subsequently used for the annotation of regions. The basis for this approach is the segmentation of image contents into regions (blobs), which are classified into region types using a variety of features. Subsequently, a training set is used to construct a table of conditional probabilities $P(w_i|b_i)$, providing the probabilities of translating a blob b_j into the word w_i, in other words, the association probability of a blob b_j with word w_i. When the association probabilities are known, the correspondences can be predicted using the EM algorithm.

However, there exist some limitations of such approaches. For effective learning a large labeled training corpus is needed, and semantically meaningful segmentation for images is in general unavailable. In addition, due to the large number of semantic classes, the mentioned approaches can be regarded as a *multi-class* classification problem [Ino04], which makes annotation a barely unsolvable task. Approaches based on region-to-word mappings (like translation or co-occurrence models) suffer from the problem of biased word distribution. If the term frequency of used words is *not uniformly distributed*, only a small number of words appear very often as annotations and most words are used only by a few images. This inaccurate co-occurrence statistics leads to a stronger association of frequent words with many irrelevant image blobs and thus degrades the annotation quality. Due to the mentioned difficulties, the accuracy of mappings between the low-level features and some high-level semantic labels (e.g. landscape, architecture, and animals) is under the requirements of annotation based image retrieval systems.

Semi-automatic Annotation

Automatic annotation avoids any users' interaction during the annotation process. However, the automatic construction of semantic knowledge from the extracted low-level features is barely possible and the derived annotation models are often afflicted with uncertainties. Hence, methods which combine automatic computation with human perception are of great interest [BS01]. For example, in [WDS$^+$01] a progressive annotation is proposed, which is embedded in the course of integrated keyword-based, content-based retrieval, and user's feedback. A probabilistic model integrating content-based techniques, statistics, and the usage of conceptual knowledge was proposed in

[CC03] to find possible keywords for a new image. Here, a semantic network is used as a representation of the relations between stored images and keywords. In addition, each keyword corresponds to a concept with a certain weight, which is adapted in the annotation/retrieval loop. The provided network of concept/keyword relations is integrated into the determination of possible keywords. As a consequence, keywords which occur in the annotation of similar images may come into consideration for the annotation of a new image, or contrariwise a keyword will be ignored if it is irrelevant for the annotation of similar images.

In general, semi-automatic annotation can be defined as an iterative process which combines keyword-based search with CBIR and user's feedback in order to suggest or refine existing annotations during retrieval. A such user feedback could be performed in the following manner: When a user poses a textual query and then evaluates the individual result tuples, keywords may be automatically added to the images that receive positive feedback. This approach can be realized by updating the terms' weights after each feedback or annotation step. When we assume, that images have been initially annotated manually and that the terms' relevance for the annotation of an image is determined by the hypothesis that similar images may share the same keywords, the annotation for an image is determined by the following steps:

1) User poses a query using either *keywords* or example image (*QbE*).

2) *if keyword:* Determination of k most similar images according to their annotations. After user's relevance feedback, the search keyword is added to positive images, if image is not annotated.
   If keyword exists in the annotation of this image, the keyword weight is increased by the positive feedback, or decreased by negative feedback.

   if QbE: For the query image I_q, the k most similar images $I_1, \ldots, I_k$ are calculated based on their low-level features. Then the user evaluates the results and gives his feedback on one or more images.

3) Analysis of frequent keywords associated with the k images, and using them for the annotation of the positive examples. Annotations are updated by removing keywords with a weight below a threshold.

The annotations can be composed of free text keywords or instances from complex ontologies which allow the fine-grained specification of objects and actions depicted in the image.

2.3.3 Summary

In this section we have presented an overview of existing annotation problems and summarized approaches which attempt to solve them. In summary, we can say that advanced semantic annotation techniques significantly improve the management and the retrieval of multimedia contents and thus have become indispensable in both professional and personal applications. The advantages of image annotation are obvious: Queries can be formulated in natural language (e.g. images of sports, mountains, and water), or even as a combination of query-by-example and natural language statements (e.g. images of buildings like this image). Although automatic approaches displace the costly and time-consuming manual annotation, they bring out some uncertainties. The semi-automatic image annotation minimizes the drawbacks of the mentioned approaches by incorporating CBIR with relevance feedback and semantics.

Having discussed the characteristics and limitations of existing annotation technologies, in the next chapter we focus on the requirements of a general semi-automatic annotation framework. Specifically, we look at the ways to capture and update the semantic knowledge needed for the image annotation and methods for efficient incorporation of users' feedback into image organization, categorization, and semantic annotation.

3

Framework for Supporting Image Annotation

Advanced annotation techniques of multimedia data significantly improve representing and retrieving multimedia-based contents. For this reason, the first contribution of this thesis is a framework for semi-automatic annotation which includes, beside components for the extraction of elementary low-level features and relevance feedback, methods for the incorporation of semantic knowledge into the retrieval process. Furthermore, the annotation component serves as an interface for the users' feedback, which is needed for an interactive construction of an annotation ontology. In such an ontology, concepts and their properties can be defined and refined at both visual and semantic level. The annotation component is tightly coupled with the retrieval component, which is responsible for the analysis of the logical structure of already annotated data. Since the projection of visual features into a finite set of semantic concepts presents a real challenge, possible solutions and approaches for this problem are presented and discussed. This framework for semi-automatic annotation is presented in [VC05b, VC05a].

3.1 Motivation

Retrieval by image content has received great attention in the last decades. Although there exist several CBIR techniques which have been presented in the previous chapter, there are still many unresolved issues in content-based retrieval systems: First, the semantic gap, which is the discrepancy between human perception and the features determined by automatic extraction algorithms, complicates queries at semantic level. Secondly, automated retrieval methods based on low-level features are mostly restricted to particular application fields causing the assignment to heterogeneous image collections to fail in terms of accuracy. The direct motivation for our work is the fact that users are highly interested in querying images at conceptual and semantic level, not only in terms of low-level features. The need of enhancement of the retrieval performance and the importance of 'semantic meaning' makes a detailed image annotation indispensable. Presently, most of the image database systems utilize manual annotation, where users assign some descriptive keywords to images. Although this process takes away the uncertainty of fully automatic annotation, it requires a high effort in exchange. Another weak point is that indexers often use different descriptors and their perceptual subjectivity may differ. In summary, since it is very difficult to automatically construct semantic knowledge from the extracted low-level features and map them on human perception, methods which combine both approaches are of great interest.

In this chapter, we present our framework for semi-automatic image annotation which combines the analysis of visual contents with the manual description of image data. Semi-automatic annotation can be defined as an iterative process which integrates keyword-based retrieval into CBIR systems and utilizes user's feedback in order to refine existing annotations or class membership of the data. The remainder of this section introduces the levels of image representation and gives the problem description of the semi-automatic annotation. Section 3.2 introduces the architecture of our system and describes the collaborations between its components. The capturing of semantic knowledge and steps required to generate concept-specific image representations are detailed in Section 3.3. Finally, we survey related work in 3.4 and subsequently, we give a summary of the presented approach.

3.1.1 Image Representation Levels

The core element of an image retrieval system is its underlying knowledge representation model. In case of image data, the *image data model* provides the basis for conceptual data representation. In the literature, several image data models and description schemes have been proposed, e.g. [GS00, SOCP99], each of them aims to provide a representation which allows to define relations between different entities and thus to capture the knowledge of a specific application domain. Furthermore, the most important requirement for the data model is its expressiveness to qualify the structure and contents of the underlying image data, the included image objects and the relations among them [GS00]. In addition, the data model should be extensible. The design of an appropriate image data model will ensure smooth navigation between images stored in a database and facilitate fast access to their logical representations. Since the image data model presents the basis for the design of an annotation/retrieval system and is tightly coupled with the system components, it should be defined first.

Image Data Model

An image object I is modeled as a composition of two layers: the *physical* and the *logical layer.*

Physical image representation $\mathcal{R}_P(I)$ is related to raw image data obtained during the image input or storage and includes the image described by a bitmap, which is stored as an array of pixel values.

Logical image representation $\mathcal{R}_L(I)$ serves as an abstraction of the physical image representation. It denotes the *feature characteristics* and *semantic information* of the image data including global image characteristics, the location and spatial relations between recognized image objects and the semantics associated with them. This information is added to image data during feature extraction and image annotation, and is highly mandatory for semantic image indexing and retrieval purposes.

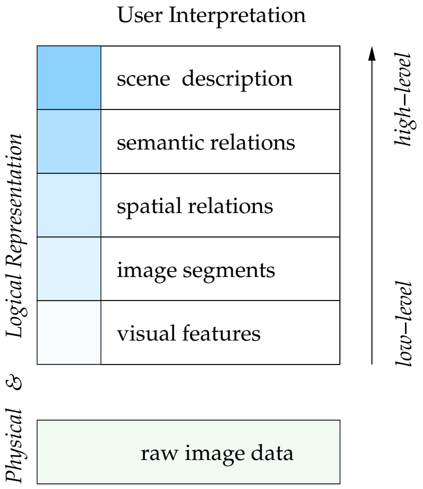


Figure 3.1: Levels of Image Representation

In order to achieve a high precision in the description and thus facilitate semantic retrieval, image contents have to be represented at multiple levels. At the bottom of the hierarchy from *low-level* to *high-level* descriptors, which is presented in Figure 3.1, an image object I is represented by a set $F_I = \{f_i\}$ of primitive visual features. For every given feature f_i, there exists a corresponding set $R_I = \{r_{ij}\}$ of representations [HROM98]. The visual features of an image are extracted by the sequential application of image processing operators to the physical representation of the image. In order to attach image regions with semantic content in subsequent steps, the image data has to be divided into information-bearing regions, the so-called *image segments.* The image segments and their spatial relationships are determined by automatic or semi-automatic segmentation methods. The transition from a set of segments to the recognition of objects presents a great challenge in the field of object recognition. Several approaches have been proposed in last years, like [AAR04, PP00], which deal with methods for automatic detection of objects in images. The disadvantages and limitations of these methods, however, is the tight coupling to a specific application field. The top-level of the model hierarchy comprises scene recognition and user interpretation. Descriptions assigned at this level usually represent abstract objects and scenes recognized by the user. User interpretation tries to describe highly subjective concepts such as feeling and emotions.

3.1.2 Problem Description

Before presenting our framework for semi-automatic image annotation, we need to precisely define the problem.

Let **D** be a database including a set of images $\mathcal{I} = \{I_1, \ldots, I_N\}$ characterized by their feature vectors $\vec{f}_{I_1} \ldots \vec{f}_{I_N}$, whereas $\vec{f} = \{\nu_1, \ldots, \nu_l\}^{\mathrm{T}}$. We are given a set of M $(M < N)$ manually annotated images that constitutes the training set $\mathcal{T}^{train} = \{t_1, t_2, \ldots, t_t\}$, where $t_i = (\vec{f}_{I_i}, \Gamma_{I_i})$ denotes the tuple of low-level features and the corresponding annotation Γ_{I_i} of image I_i. Then, the annotation problem can be described as follows:

Problem 3.1 *Annotation Problem.*
Given an unlabeled image I_q characterized by its low-level features $\vec{f}_{I_q}$, use the training image set $\mathcal{T}^{train}$ to predict an 'accurate' set of keywords $\mathcal{K}_{I_q} = \{k_1^q, \ldots, k_m^q\}$ (or the annotation Γ_{I_q}) which effectively describes the content of I_q.

Generally speaking, we are looking for a function f^{AN} from the image's physical representation $\mathcal{R}_P(I_q)$ to its logical representation $\mathcal{R}_L(I_q)$:

$$f^{AN} : \mathcal{R}_P(I_q) \rightarrow \mathcal{R}_L(I_q). \tag{3.1}$$

In order to facilitate high-level retrieval from image databases, the image data should be interpreted and annotated when it is inserted into the database.

3.2 Architecture

The principal objective of our annotation system is to provide users an image retrieval system with the capacity to evaluate image classification and assignment of the data to high level concepts. The basic feature of the annotation system is the manual association of the image data with descriptors from existing ontologies. Furthermore, by analyzing the logical structure of already annotated images, the system provides a semi-automatic annotation which generates descriptions for a new and unlabeled image and thus proposes the membership of this piece of data to a predefined category. Since the system is working *semi-automatically*, it depends on an interactive user's feedback at several processing steps. Figure 3.2 illustrates the components of our $\mathcal{IKONA}$[1] annotation system:

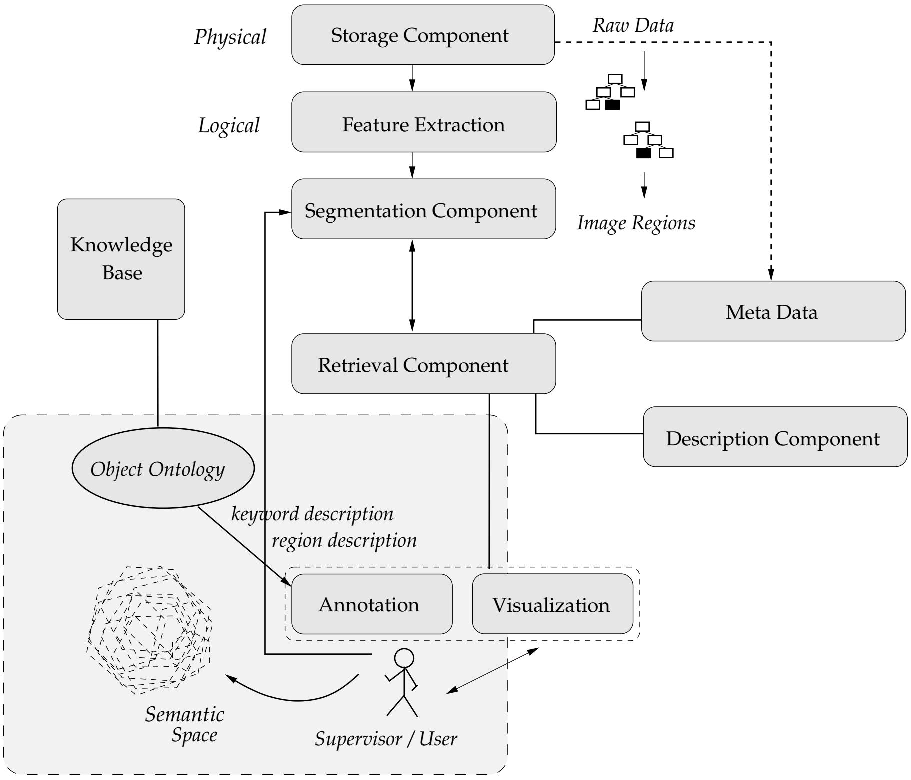


Figure 3.2: Architecture of the Image Annotation Framework $\mathcal{IKONA}$

Visualization Component. This component consists of an *image data display* and a *summarization display*, which generates thumbs from a subset of images belonging to one category or returned as query results. Furthermore, the graphical user interface provides a visualization of the *semantic knowledge* (semantic concept space) used for partitioning the image data into a set of semantic concepts. Additionally, it serves for an exemplification of retrieval results and the features considered for relevance computation.

Retrieval Component. This component controls the retrieval process. Beginning with *query formulation* and its *interpretation*, which is performed by parsing and compiling of the query into an internal format, the component provides functions

[1] $\mathcal{IKONA}$ (greek origin): image, figure

for similarity computation between the query object and the underlying data stored in the database.

Feature Extraction Component. Beside the determination of basic *meta data* (e.g. date, creator, or filename) from images, this component mainly provides methods for extracting primitive (visual) characteristics of images. For example, the set of *low-level* features implemented in $\mathcal{IKONA}$ system currently includes color features (like color statistics and color histograms), color moments, and texture characteristics.

Segmentation Component. In order to find out the semantic relations between words and 'objects' contained in an image, it should be divided into objects. For that purpose, an automatic segmentation algorithm based on the *region growing* approach [AB94] is provided by the *segmentation component*. Since this segmentation approach is based on low-level homogeneity criterion such as color and texture, it remains essential to involve user's perception and provide an interface for manual segmentation of image regions. Through the interactive segmentation user-interested regions can be emphasized.

Description Component. Content descriptions of the images are stored in a relational database. This component provides methods for *description matching* which are used to compute the overall similarity between the content description of a query image and the content descriptions of images in our collection.

Semantic Concept Space. The C-dimensional concept space results from a projection of the image feature space into a variable set of concepts from the object ontology. This concept space serves as a user's representation for his own view of the image collection and provides information of concepts, their properties, and weighted relationships to other concepts in the application domain. Furthermore, for each concept there exits a suitable *semantic annotation template* which serves as a template for the semantic description for a given concept.

Annotation Component. The annotation component provides an interface for attaching images with semantic descriptions which is done through annotation templates stored in the semantic concept space and providing users entries for image description with a structured set of features at both feature level and semantic level.

Object Ontology. For different application domains an object ontology is created in order to provide a formal specification of concepts and their relations. The concepts are taxonomically arranged, which testifies their relations and thus allows automatic inferences on knowledge for extended annotation. This ontology is a subset of the **Knowledge Base**, which represents an abstract model for the semantic knowledge.

Basic Metadata. Metadata contains standard information of the image raw data, like date, the photographers name, or the filename.

Storage Component. The storage component represents the hardware of our system. It encapsulates the physical data items stored in the database from components responsible for the analysis of the image features or the extraction of the associated metadata.

3.3 Capturing Semantic Knowledge

This section concentrates on the components within the dashed box of the architecture illustrated in Figure 3.2, comprising *visualization*, *annotation*, *semantic space*, and the *object ontology.* The *semantic annotation template* (SAT) should both provide an understandable schema for attaching semantic meaning to images and simultaneously serve as knowledge acquisition interface.

3.3.1 Schema and Generation of Annotation Templates

An annotation template is generated dynamically through the combination of knowledge (object ontology), automated feature extraction, and relevance feedback. Its unique structure is configured for each class of concept entities. Templates for subclasses are generated by inheritance of the structure of the class template and by adding specialized descriptors. The template comprises the following description fields (see Figure 3.3):

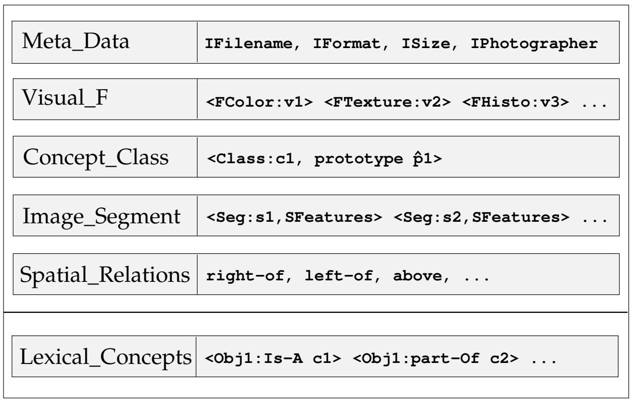
Metadata. The permanent information about an image object is provided by its metadata, like `filename`, `format`, `size` or the `photograph ID`. This information is unique for an image and can be easily extracted.

Basic Keywords. Keywords are features describing high-level domain concepts which appear on several abstraction levels [GRV96]: In the first instance, the visual appearance and structure of the image contents is described in terms of regions and their spatial relations. For that purpose, the image is partitioned – automatically or manually – into n content bearing segments comprising objects including their type, identity, and other properties, like activity, event, etc. Then, the corresponding annotation Γ_I of an image I consists of a set of keywords from the set $\mathcal{K} = \{k_i, k_2, \ldots, k_m\}$ ordered by their probability of being adequate as descriptors for I. The selection of further keywords does not depend on the presence of visual concepts in the image, but rather specifies the meaning of image contents recognized by humans in form of implicit descriptors (*imdescriptors*).

Visual Features. For each concept the prototype vector $\hat{p}$, which will be defined in Section 3.3.2, with its weighted feature components is represented to the user. During relevance feedback, the user indirectly controls and specifies its feature weights which are computed on the basis of member images of a given 'visual' concept.

Semantic Knowledge. There are several entries for semantic knowledge in the template. At first, either the user can specify the concept class of an image or it can be automatically determined during the alternating retrieval and annotation steps. In addition, the template includes derived semantic relations between concepts and logically inherited attributes from super-concepts. In order to facilitate retrieval at the semantic level, assigned keywords are associated with a terms coming from a thesaurus providing noun relations (like `Is-A`, `Part-Of`, `Synonym`) or causal relations between entities. This means, that keywords are part of a hierarchy and can be both utilized to expand the query by following the semantic relations and to serve as an additional dictionary to propose alternative keywords for image annotation.

Implicit Information. Knowledge about the image contents implicitly defined by the user can be recorded in separate description fields. This knowledge comprises the `emotions`, `movements`, `time`, `place` and `activities` of entity objects.



Meta_Data	IFilename, IFormat, ISize, IPhotographer
Visual_F	<FColor:v1> <FTexture:v2> <FHisto:v3> ...
Concept_Class	<Class:c1, prototype $\hat{p}$1>
Image_Segment	<Seg:s1,SFeatures> <Seg:s2,SFeatures> ...
Spatial_Relations	right-of, left-of, above, ...
Lexical_Concepts	<Obj1:Is-A c1> <Obj1:part-Of c2> ...

Figure 3.3: General *Annotation Template* for a Concept C_i

3.3.2 Mapping Visual Features to Semantic Concepts

In order to classify images to semantic concepts, they have to be clustered at the feature level first. In our approach the initial semantic categories of images are specified by unsupervised learning using the *k-Means* clustering algorithm. The basic idea of k-means is to find k mean vectors $\mu_1, \ldots, \mu_k$ (or *centroids*), one for each cluster, so that the total intra-cluster variance, and thus the sum of squared error E will be minimized:

$$E = \sum_{i=1}^{k} \sum_{j=1}^{N_j} ||x_{ij} - \mu_i||^2 \,, \tag{3.2}$$

where x_{ij} represents the j-th point in the i-th cluster, μ_i is the centroid of the i-th cluster, and N_j denotes the number of elements assigned to the j-th cluster.

In general, the k-Means clustering works as follows:

1. Initialization of the centroids by partitioning the input points into k initial sets, either randomly or using some heuristic data,

2. For each data point, the membership to a cluster is determined by choosing the nearest centroid (e.g. by Hamming distance or Euclidean distance),

3. Computation of new centroids $\mu_1, \ldots, \mu_k$ for the new clusters ,

4. The steps 2 and 3 are repeated until convergence, which is obtained when the points no longer switch clusters, or centroids are no longer changed.

Such a cluster centroid can be regarded as the representative vector $\hat{p}^{c_i}$ for the cluster c_i and can be used to represent a 'visual' concept in our image collection. Since image retrieval deals with high-dimensional data characterized by both a large number of attributes (or features) and with noise, clusters are often hidden in subspaces of the data and do not show up in the full dimensional space. For this case, methods like *subspace clustering* [AGGR98] aim at detecting clusters in any subspaces of the original feature space and additionally serve as dimension reduction.

In our approach we used a modification of subspace clustering combined with feature weighting to identify and characterize semantic clusters embedded in subspaces. This method allows us to identify only those features which describe best a particular class of images and thus facilitate a better separation of the corresponding data points than in the original space. This vector can be considered to accurately represent overall characteristics of the images that belong to the same category. Let $\hat{p}^{c_j} = \{\pi_1, \ldots, \pi_l\}$ be the prototype vector representing the cluster c_j, then its i-th component is computed by

$$\pi_i = \frac{1}{|c_j|} \sum_{I \in c_j} \nu_i(I), \tag{3.3}$$

where $\nu_i(I)$ denotes the i-th feature component of image $I \in c_j$ and $|c_j|$ denotes the number of images belonging to category c_j. The prototype vectors have the same dimensions as the feature vectors of the images. To perform a selection of a subspace from the feature components, a weighting of the components relevant for the distinction between other categories is needed. As a general rule, local and global criteria are combined for weighting. Let the image database consist of N images, and let ν_i be one of the feature components that is essentially for a category of images or for a class c_m. The weighting w_i of the i-th component of the prototype vector $\hat{p}$ is computed as follows:

$$w_i = \text{freq}(\nu_i, c_m) \log\Big(\frac{N}{\text{occ}(\nu_i, \mathcal{C}')}\Big) , \tag{3.4}$$

where the feature frequency $\text{freq}(f_i, c_m)$ represents the occurrence of feature f_i in images assigned to class c_m and $\text{occ}(f_i, \mathcal{C}')$ denotes the occurrence of this feature f_i within other classes $\mathcal{C}' = \mathcal{C} \setminus c_m = \{c_1, ..., c_{m-1}, c_{m+1}, ..., c_n\}$. In order to not recalculate these occurrences we use a matrix $\mathbf{M}$ which describes the occurrences of features in concept classes; it is a sparse matrix whose rows correspond to classes and whose columns correspond to features.

3.3.3 Relevance Feedback at Semantic Level

As mentioned in Chapter 2, the relevance feedback technique tries to bridge the gap between low-level features and high-level semantics in retrieval systems and is achieved by users' interaction with the system. Usually, users evaluate the individual result tuples and according to this, the system reformulate the query to better reflect the information need.

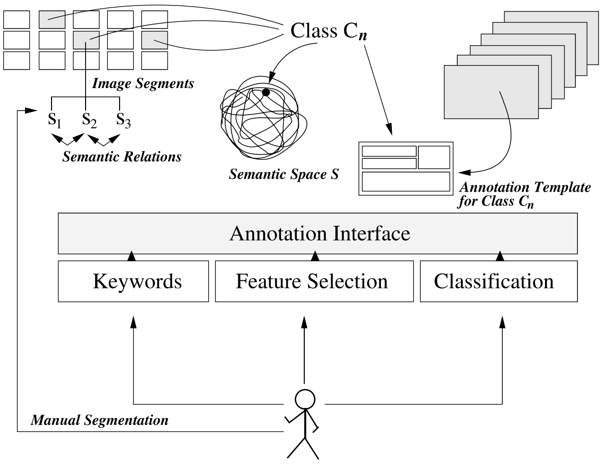


Figure 3.4: Relevance Feedback at Multiple Levels

The multiple levels of relevance feedback and their integration into our annotation framework are summarized in Figure 3.4.

Initial Point

At the beginning, where no annotation has been made by users, the images are unsupervised clustered (e.g. by using the k-Means clustering algorithm) due to their low-level feature similarity. Until then, the annotation template for an image is determined by taking the annotation of that image cluster (class) with the largest similarity to this image. The first training examples are provided by the user's annotation and are used to learn an annotation template for a specific concept class. The properties of a concept class and their subclasses are refined iteratively during the alternating retrieval and annotation steps.

Relevance Feedback during Searching

When user submits a query consisting of a keyword or a concept from the ontology, the query string is automatically assigned to the image's annotation. Depending on the class and user-selected image segments, which are again associated with concepts, images with similar semantics are automatically grouped in the immediate neighborhood in the semantic space. Specializations of semantic classes are recorded with their distinguishable low-level features and the resulting modifications in the description fields of the template. The user can refine results by using negative and positive examples and update the knowledge about image classes in the semantic space. Each time a feedback or a new annotation is provided by the user, the following data has to be recalculated:

a) prototype vector $\hat{p}$,

b) semantic space is updated by adding concepts with relationships to other concepts,

c) semantic template for this image class can be refined.

Relevance Feedback during Annotation

In the next step, the generated templates have to be linked at semantic level in the Semantic Space. Firstly, the prototype vector $\hat{p}$ is adjusted by assigning similar images to the present image class. Until now, our semantic space only consists of a set of disjoint concept classes $c_1, \ldots, c_n$ and their low-level characteristics. In the annotation process we use relevance feedback procedure to define rules for mapping images classes to a semantic annotation template. In addition to the known correspondences between concepts and visual low-level features, a set of rules is constructed to map concepts to a semantic template and finally map concepts to a controlled vocabulary.

Through relevance feedback from users the semantic knowledge for an appropriate description template is accumulated, which gradually enhances the annotation process. Traditionally, relevance feedback techniques proposed in the literature operate on the low-level features such as color, texture, or shape and are based on modifying search parameters as to better represent the *concept* the user is looking for. For these purposes relevance values (e.g. negative or positive) can be assigned by the user to all retrieved images, which leads to a modification of the query vector (query point movement) or adaptation of the similarity metrics [LHZ+00]. In the annotation approach, this relevance values can be supplemented with placing the same or similar keywords (concepts) to a set of images.

In heterogeneous image libraries however, images of the same concept class are not likely agglomerated in the selected feature space. To this end, semantic-based retrieval and clustering demand computations in a subspace in which the concept class lies [ZH03]. For example, an image of a 'black dog' in the low-level feature space is not necessarily closer to a 'white dog' than it is to a 'black tiger', if the discriminating feature is color.

In addition to the query point movement, a re-weighting at the concept-level has to be performed. The weights of the representative features of the prototype vector $\hat{p}$ have to be updated and the semantic space has to be reorganized.

Creation of User Profiles through the Feedback

During the alternating search and annotation steps a *user profile* is created, which consists of his own object ontology used for the annotation (*annotation ontology*) and a set $\mathcal{L}^u = \{l_1^u, l_2^u, \dots, l_m^u\}$ of user-specific contexts. A certain user context $l^u(q)$ for a query q is defined by a set of concepts and an optional set of negative constraints, which comprises keywords or concepts to be excluded in a query. These constraints are selected by the user during the interaction with the system. Based on user behavior, a specific context in the user profile can be updated or a new context can be added. Such a user profile is utilized to provide the user with his/her own annotation ontology that is more consistent with their view of the world and can be used for a query expansion according to the user's interests. The application of such user profiles is detailed in Chapter 5.2.

3.4 Related Work

In recent publications, the research has focused on approaches for automatic image annotation like presented in [JLM03, PYDF04]. These approaches are based on discovering correlations between image features and keywords, which are subsequently used to estimate the probability that a given term is suitable for the description of an image region. An architecture for semi-automatic image annotation has been also proposed in [WDS$^+$01], which integrates keyword-based search, content-based image retrieval, and user feedback. This approach is presented from the perspective of enriching the image data by keywords, which are extracted by considering results of the feedback cycle performed by the user. The initial search keywords are automatically added to the images that receive positive feedback and facilitate keyword-based image retrieval in the next iteration. A probabilistic model was proposed in [CC03], which integrates content-based techniques, statistics, and the usage of conceptual knowledge in order to find possible keywords for an unlabeled image. Here, a semantic network is used as a representation of the relations between stored images and keywords. In addition, each keyword corresponds to a concept with a certain weight, which is adapted in the annotation/retrieval loop. The provided network of concept/keyword relations is integrated into the determination of possible keywords. As a consequence, keywords which do not occur in the annotation of similar images may come into consideration for the annotation of a new image, or contrariwise a keyword will be ignored if it is irrelevant for the description of similar images. Another tool for semi-automatically annotating image regions is presented in [PM95], which is based on manual selection of positive and negative examples and then uses texture similarity to propagate annotations. In several papers, the choice of appropriate annotation terms is supported by existing ontologies [HSWW03]. We also found several variants of relevance feedback [PMS96, KK93] using learning methods and model inference to find correspondences between the high-level concepts users perceive and the low-level features extracted from the images. Several approaches in the area of semantic information retrieval incorporating *mappings* of local features into words have been proposed [Lim01, LTM03]. These approaches are based on the creation of a partial taxonomy for home photos, modeling of high-level information like events, and the definition of visual keywords to describe semantic concepts.

3.5 Summary

In this chapter we have introduced a framework based on multi-level relevance feedback for semi-automatically annotating image collections. The proposed framework includes, besides components for the extraction of low-level features, methods for the incorporation of semantic knowledge into the retrieval process. The annotation component is tightly coupled with the retrieval component, which is responsible for the analysis of already annotated data. Since the projection of visual features into a finite set of semantic concepts presents a real challenge, a clustering algorithm supplemented by the weighting of feature components has been presented in order to represent 'visual' clusters of the image collection. In conclusion, through the iterative retrieval and annotation process on both low-level features and high-level semantics, labeled training data and knowledge needed for clustering and annotation is obtained. The resulting semantic knowledge can be embedded into image retrieval systems and helps users to keep track of the underlying image collection. But in the first instance, the semi-automatic annotation improves the exhausting manual image annotation without the uncertainty of fully automatic annotation. Of course, the performance of semi-automatic annotation depends on the performance of the CBIR algorithms, but generally speaking we can accept this drawback for the retrieval accuracy semantic annotation provides.

4

Unifying Different Users Interpretations and Levels of Abstraction in Image Retrieval

Assigning annotations still remains indispensable in both professional and personal retrieval applications because they facilitate textual or conceptual queries in large image repositories and thus classify the image data into semantic categories. However, different users' perception of image contents and the lack of standards among different annotation tools make it necessary to develop methods for the unification and integration of different annotation schemes. In this chapter we summarize the problems occurring during the annotation process and present a representation technique for the complex semantic annotation space which results from the transformation of the subjective perceptions into a unified knowledge base. Our technique is used to bridge the discrepancy between users' vocabulary and the several levels of abstraction at which content descriptions are assigned. Based on examples, we show how to integrate our method into probabilistic approaches for (semi-) automatic image annotation.

4.1 Motivation

The representation of semantics has been identified as being crucial for facilitating intelligent search and retrieval from multimedia databases. In our work, annotations are used for the conceptualization of multimedia data (e.g. images, videos, texts, etc.) and are understood as an accumulation of strongly personalized information given by users which have different standards of knowledge and act in different contexts. When annotating an image, the user conceptualizes and describes the data content by capturing all or some *objects* in various levels of detail, for example people, scenes, actions, etc. The human understanding of the data contents is given by the natural capability to immediately interpret, categorize, and identify interrelationships in the data. But the user's subjectivity may appear at several points, for example at the querying step in form of users' preferences and skills or through the differing background knowledge during the annotation [Ino04]. Since this form of *information overload* complicates the search and makes the retrieval of relevant information an exhausting task, a formal framework is needed to represent the knowledge in a human and machine-understandable way, both for the automatic analysis of raw multimedia content and the extraction of the given semantic annotations. In addition, another important requirement for annotation-based systems is the flexibility to accommodate differing semantic views of the same image and the dynamics to handle the advances in the areas of image processing as well as the evolution of application domains [GMY93]. Furthermore, it is desirable that an image retrieval system will be able to adapt itself continuously to the changing requests of the user [PMS96] by adjusting the changing mappings between image data and its annotations (e.g. by relevance feedback).

The new idea in our approach is to integrate hierarchical multi-level information that is contained in annotations into an image annotation and retrieval framework. In this context, *'multi-level'* means that annotations are not considered as independent keywords, but rather as descriptions which are assigned at multiple levels of abstraction, visually structured at *object level* and semantically structured at *description level.* Semantics is commonly defined as the meaning of data, and the task of evaluating the extent of semantic matching between different annotations should be based on their meanings. Since in the most cases the *meaning* of a piece of data cannot be expressed by only one concept, we introduce a graph-based representation technique for annotations which encodes the semantic relations between images, and organize them in a human and machine-understandable way. Our method incorporates semantic relations between annotation terms, like specialization or syntactic relations, and thus facilitates semantic retrieval at different levels of abstraction. By introducing the relevance $H[C_i, k_j, l]$, denoting the importance of a keyword k_j for the description of the concept C_i depending on a given context l, we can determine cluster of images with a frequent occurrence of this keyword in the annotation space and thus discover its relations to other annotations.

This chapter is structured as follows: In Section 4.2 we briefly review the properties of the *Image Annotation Process* and the encountered problems, like users' subjectivity. In addition, basic definitions are introduced. In Section 4.3, after the description of our semantic model for annotations, the *Annotation Analysis Framework* with its functionalities for analyzing and encoding different abstraction levels in annotations and the graph-based representation for multi-level annotations is presented. Afterwards in Section 4.4, we demonstrate the application of the resulting annotation space for the probabilistic image annotation. A summary of related work is given in Section 4.5. Finally, Section 4.6 concludes our approach and gives further research directions.

4.2 Image Annotation

Users' interpretations can be summarized by means of *terms* or *keywords* describing the recognized semantic concepts. The association of these keywords with images for capturing their semantic contents and enriching them by additional information is known as *Image Annotation*. At the same time, the annotation should assign the image data to one or more of the predefined categories resulting in a semantic classification of the underlying data collection. Ambiguous interpretations can be avoided by using a lexicon-based knowledge (e.g. an ontology) which serves as a source of semantic types and their relations. In order to combine the high-level tasks of *scene recognition* and *user interpretation* with traditional CBIR systems, the manual annotation is performed by users. Figure 4.1 illustrates a course of image annotation according to human perception ability and the corresponding image data model used for modeling content information. Accordingly, the image annotation process includes the following steps:

1. Applying visual analysis of the image contents in order to identify relevant objects or regions and their relations.

2. Determining a set of candidate keywords for the annotation of the image by using an application-specific lexicon. These textual keywords are supplemented by attribute based meta data, such as creator, date, genre, file type, size, etc.

3. Assigning a set of keywords to the image at different abstraction levels, for example by describing the recognized objects, their relations, and the overall classification of the scene. To perform clustering at semantic level, information about the low level features, like color, texture, and (primitive) shape within the image has to be associated with the recognized semantic concepts.

4.2.1 Problems occurring during Image Annotation

Although the annotation process appears to be a straightforward task which seems to succeed error-free, it is afflicted with uncertainties. Beginning with the selection of an appropriate set of keywords and the abstraction level, it turns out to be a complex task. Particularly, to make manual annotations reusable and integrate them into

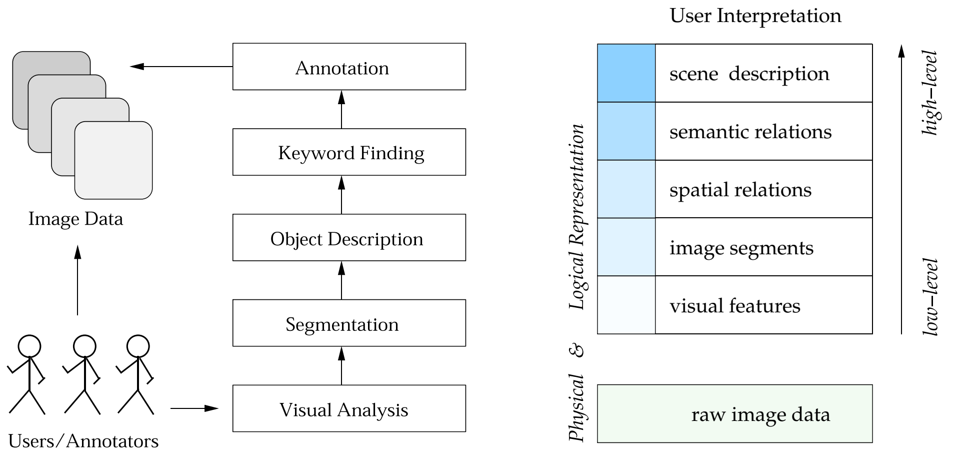


Figure 4.1: The Workflow of Semantic Annotation and the Image Data Model

semi-automatic annotation and retrieval systems, more than incoherent keyword descriptions are needed. The mostly encountered problems during the annotation process are [VC06]:

Multiple Levels of Abstraction. Annotations are assigned by different users in different contexts and from different points of view. In addition, the type of information and levels of abstraction may often depend on the application domain. Some annotations may work well with one application, but by exchanging the context they may turn out to be useless or unsuitable for reuse.

Incompleteness. Retrieval systems using semi-automatic annotation are mostly based on a supervised learning technique that compares image similarity at low-level and then annotates images by propagating terms over the most similar images. Such an approach relies on reasonable and adequate annotations which may be used as training data. The incompleteness of annotation data and the biased assignment of annotations will lead to a small recall value in search results.

Non-uniform Word Distribution and Word Sparseness. The term frequency of used words for the annotation is not uniformly distributed. Only a small number of words appears very often in annotations and most words are used only by a few images. Consequently, word co-occurrence frequencies within a set of annotated images cannot be determined. The problem of word sparseness can be overcome by incorporating additional knowledge such as annotation ontologies that explicitly identify the relationships between words and their meanings.

Hard-to-Describe Objects. Complex and hard-to-describe objects in images or objects occluding other objects can complicate semantic annotation. The extraction of semantic concepts is difficult because images may contain multiple semantic concepts and different objects corresponding to different concepts. In addition, images differ from each other in the number of objects resulting in different-sized annotations for the same semantic category of images.

Users' Perception. Users' perception proves to be highly subjective and leads to inconsistent annotations among indexers. In addition, users' views may change over time, that means that different interpretations could be assigned to the same images or the same annotations could be given to different image contents.

Subjectivity in Image Annotations

Variations in user's contextual knowledge, resulting in an unsteady quality and preciseness of content descriptions, lead to problems when retrieval is performed on annotations. This fact is demonstrated in Figure 4.2 by means of two annotations Γ_1 and Γ_2 which have been assigned by two different users to an image illustrating a building which is surrounded by greenery. The first annotation Γ_1 is a *flat* annotation composed of keywords which are not semantically related. When all or some keywords are linked to an existing ontology, mirroring their semantic relations, the annotation is characterized as *semantically (partially) structured*. For the calculation of the similarity between two structured annotations, the annotation ontology, the keyword types, and their relations have to be considered. The excerpt of the ontology (Figure 4.2, right) describing the concept 'building' (B) with its subconcepts 'university' (U), 'library' (L), 'school' (S), and 'museum' (M) and its superconcept 'city' (C) makes clear that the used keywords are related to each other and require specific rules to compute the extent to which they share similar semantic contexts.

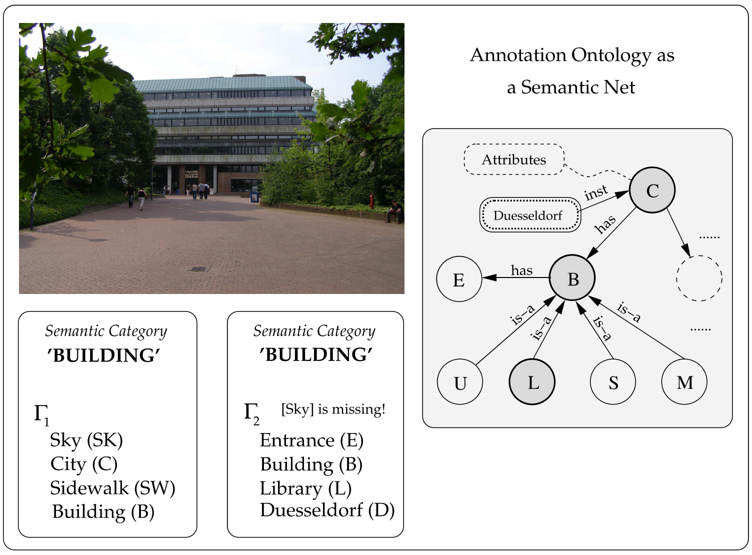


Figure 4.2: Annotations Γ_1 and Γ_2 and the corresponding Annotation Ontology

Furthermore, the subjectivity in annotations also provides advantages, because they contain contextual information derived from the annotators' view on the images. Although this subjectivity might cause some mismatches between the users' intentions and retrieval system behavior, such contextual information embedded in annotations is sometimes useful for interpreting images. For the most part, subjective context is accessible only by the annotation words assigned to images (for example keywords *'laughing'* and *'children'*). Thus, subjectivity may enhance semantic retrieval when there exist methods to understand and interpret their characteristics.

4.2.2 Definitions

The formulation of basic definitions is an essential step for introducing the Annotation Analysis Framework, which can serve as a solid foundation for the theory of annotation-based image retrieval using high-level semantics.

Semantic Concepts and Categories. We define a set $\Phi = \{C_1, C_2, \ldots, C_n\}$ of semantic concepts arranged in a concept hierarchy. The subset relation $\subseteq_\Phi$ between two concepts $(C_i, C_j) \in \Phi \times \Phi$ is a partial order between concepts $(C_i \subseteq_\Phi C_j)$, which denotes that C_i is a sub-concept of C_j. The set of the concepts is not known apriori and is dynamically extended by the user according to the appearance of a new instance of semantic concept. Images containing particular visual concepts $\Phi_\mathcal{S} \subseteq \Phi$ can be summarized into a semantic categories from the predefined set $\mathcal{S} = \{S_1, \ldots, S_t\}$. The number of sematic categories is not fixed, and is expanded during annotation and retrieval.

Representative Features. Let $\mathcal{D} = \{d_1, d_2 \ldots d_w\}$ be a set of application domains and $\mathcal{F}_{d_i}$ a set of representative visual features for a domain d_i.

Image Data Set. A database $\mathbf{D}$ includes a set of images $\mathcal{I} = \{I_1 \ldots, I_N\}$ which are characterized by their feature vectors $\vec{f}_{I_1} \ldots \vec{f}_{I_N}$.

Segmentation Set. Let $\mathcal{R}$ be the set of manually or automatically segmented *regions of interest* (ROIs). We define a function $\pi_\mathcal{R} : \mathcal{D} \to 2^\mathcal{R}$ so that $\pi_\mathcal{R}(d_i)$ is the set of representative regions of interest of a domain d_i. Thus, an image I belonging to an application domain d_i may be divided into a set of ROIs $\mathcal{R}(I) \subset \pi_\mathcal{R}(d_i)$.

Image Annotations. Let be $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$ a set of keywords. The subset $\mathcal{K}^{d_i} \subseteq \mathcal{K}$ is a set of p keywords or semantic labels $\{k_1^{d_i}, k_2^{d_i}, \ldots, k_p^{d_i}\}$ which are used in an application domain $d_i \in \mathcal{D}$. An annotation Γ_I of an image I (from the application domain d_i) is given by a set of keywords from $\mathcal{K}^{d_i}$ characterizing the content of I. The set of images attached with the annotation Γ is denoted by $\mathcal{N}(\Gamma)$ and their number is presented as $\|\Gamma\|$.

Annotation Mapping. Let $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_z\}$ be annotations which are used to describe the set of images $\{I_1, I_2, \ldots, I_z\} \subset \mathcal{I}$. Than the mapping into the *Annotation Space* is created by arranging the annotations in a multi-graph structure consisting of a set of nodes $V_\Gamma = \{\Gamma_1, \ldots, \Gamma_z\}$ corresponding to image annotations and a set of edges $e_i \in E$ ($E \subset V_\Gamma \times V_\Gamma$) connecting them. The edges are determined by the application of specific rules characterizing the semantic relationship between annotations (detailed in Section 4.3.3).

Domain-dependent Annotation Ontology. An ontology O_{d_i} provides a collection of concepts from Φ in a specific domain d_i, and their interrelationships (e.g. *is-a*, *instance-of*, *part-of*).

Representative Features	color, histogram, texture features
Semantic Category	beach images, historic photographs, sightseeing
Annotation	textual description, e.g. city, building, *London*
Image Segments	Segmentation of an image into information-bearing contents e.g. extracting objects from background
Application Domain	medical, geographic, face detection, cell detection
Annotation Ontology	conceptualization of objects and their relations, for example entities like '*library* `is-a` *bulding*'

Table 4.1: Examples

Table 4.1 summarizes the possible instances of the introduced conceptualization. Let us consider the image $I_1 \in \mathbf{D}$ in Figure 4.2, which is represented by an l-dimensional feature vector $\vec{f_{I_1}}$. The selection of features from the set $\mathcal{F}_{d_i}$ depends on the membership of the image to an application domain d_i. An image may belong to more than one application domain (for example, image I_1 could belong to *'landscape'* and to *'geographic'*). An *application domain d_i* restricts the objectives and demands on CBIR methods for the detection of particular patterns in images, in the case of geographic images algorithms for the detection of semantic concepts like 'sky', 'building' or 'tree' are needed.

4.3 Multi-level Annotations

4.3.1 Semantic Model for Annotations

The main objective of this work is to extract and unify the information from Multi-level Annotations (MLA). In order to fulfil the mentioned requirements, annotations are not only considered as a collection of semantically independent keywords. For this purpose, we introduce a general multi-level annotation structure [VC08b], which is

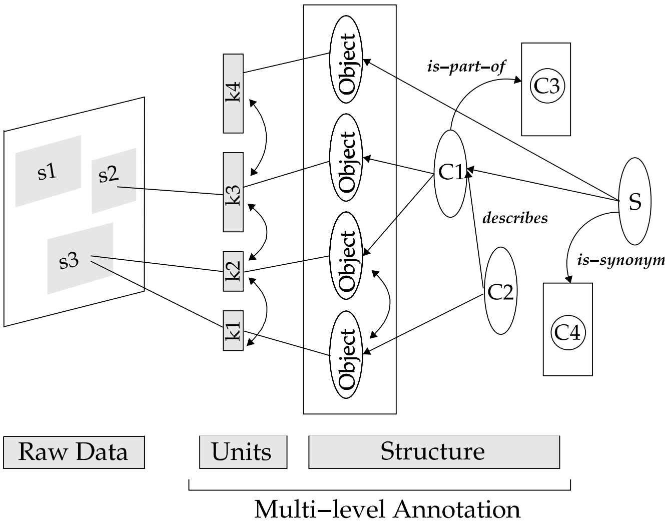


Figure 4.3: Example of the Multi-level Structure of Annotations

presented in Figure 4.3. According to it, an annotation consists of *annotation units* and its *structure* which reflects the composition and relations between the annotations units. The annotations consist of textual descriptions (descriptors) which are either linked to a part of the image data (*segment*) or unattached (implicit descriptors). The structural information consists of an *object layer* and a *description layer*. At object layer, *annotation relations* describe the 'visual' relations between annotations, e.g. the position of an object, whereas at description layer, annotations are linked to each other or to other objects, for example to feature an optional description for the same content or to describe other relational properties. The annotations' properties of reusability and generality are warranted by their flexible structure: *annotation types* define the kind of content held by annotations (e.g. *object*, *action* or *event*). A type possesses a name and the types of possibly connected annotations. Further information about an annotation relation is specified by a *relation type* which describes the type of the objects associated by a relation and defines the types and the number of participating annotations. For example, the type of *action* represents that an object invokes operations on other objects.

Relations between Keywords

Relations between annotations are needed to describe the content at multiple levels and to create structured and consistent annotations. During the annotation process, the user either defines relations between keywords according to the relation catalog (an extract is shown in Figure 4.4) or if available the relational information is extracted from the annotation ontology, which is used to define semantic and lexical relations when they cannot be inferred automatically from the image's content.

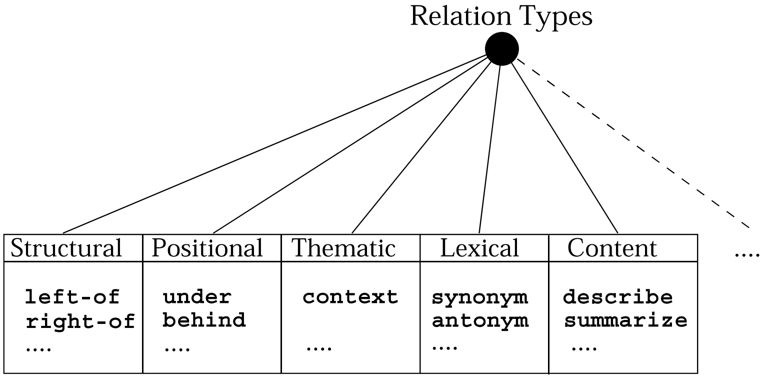


Figure 4.4: Annotation Relations at Different Semantic Levels

A relation is composed of a *relation id*, its *type* and the *levels* describing the relative hierarchical positions of two participating annotations terms. For example `left-of` describes a structural relation which denotes the spatial arrangement of two objects both at the bottom level in our image representation model (see Figure 3.1). The level declares that the arrangement of objects has been determined using low-level features (e.g. by segmentation). At a higher semantic level there exist positional relations, like **under** or **behind**, whose perception is more influenced by the user. Thematic relations, which represent a subgroup of semantic relations, connect verbal concepts with nominal concepts preferably occurring as their complements. For example, the verbal concept *write* should have pointers to the concept *person*. Another relations, e.g. lexical, are used to mitigate synonymy and polysemy problems in the retrieval process. By providing such a finite catalog, the possible relations between concepts are constrained, a fact which reduces the amount of annotation errors and moreover simplifies the analysis of the relations. By the way, the inference making process can be used to discover hidden relationships.

4.3.2 Components of the Annotation Analysis Framework

Since in the majority of cases the application domain in which annotations will be used in the future is unknown at the annotation time, methods for their understanding and interpreting are required. The development of an *Annotation Analysis Framework* is an essential step to the unification and integration of different annotation schemes. The thus resulting annotations provide a semantically consistent description of the data which will result in a higher precision and recall in image retrieval. For this purpose, a statistical approach combined with lexical analysis is used to find correspondences between the used keywords and visual concepts, or to find the most frequently used annotations for a particular 'visual' concept.

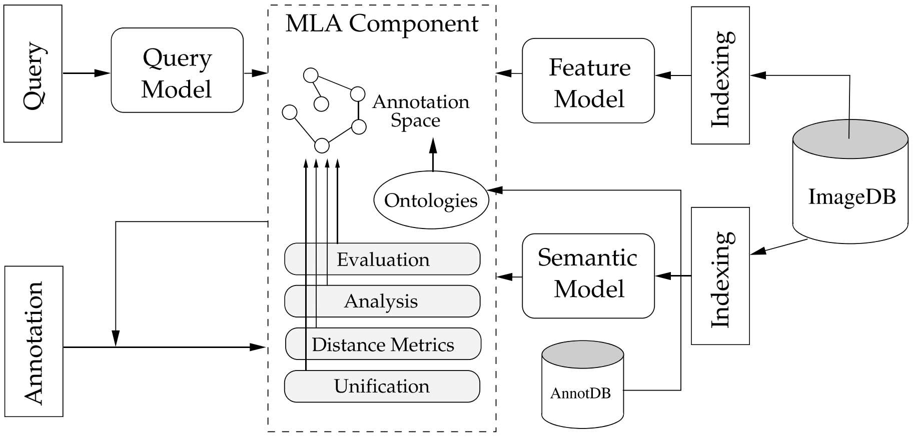


Figure 4.5: Annotation Analysis Framework

The four main functionalities of the Annotation Analysis Framework, demonstrated in Figure 4.5 are the following:

Unification of Annotations and Inference. The unification of annotations which have been inconsistently created by different users and the determination of redundant information is done. Furthermore, semantic inference rules (extraction of relationships between concepts) can be used to derive new knowledge from the existing annotation ontology.

Annotation Analysis. By considering already annotated data annotations assigned to a particular concept are analyzed. Measures like *preciseness* and *visual expressiveness* describe the quality of an annotation, which is helpful to determine a suitable abstraction level or the optimal length of an annotation.

Context-based Distance Functions. Distance functions should take into consideration the different views and relations between annotations and the context they appear in. This aspect is introduced in Chapter 5.

Statistical Evaluation. The evaluation of annotation co-occurrences allows query expansion preventing that different users' views negatively influence the retrieval results. Moreover, by using associations between image's low-level data (features) and the assigned keywords an appropriate propagation of new annotations can be performed.

Example 4.1 *Unification of Annotations and Inference.*
In the first place, the conceptual distance between a set of image descriptions (e.g. annotation A) and a structured global annotation ontology is computed in order to determine the amount of information they share. Secondly, the unification is done by

finding a covering of the annotation terms with the given ontology. Assuming that we have

- Annotation A: {*skyscrapers, New York*},
- Ontology with concepts 'city' and 'building',
- Relations, e.g. (*skyscraper* **is-a** 'building') and ('building' **is-part** 'city'),

then the rule (*skyscraper* **is-part** 'city') can be inferred. The unification provides an annotation $\bar{A}$: {→building:*skyscrapers*, →city.inst:*New York*} with pointers (→) to the respective concepts in the ontology.

4.3.3 Graph Representation for Multi-level Annotations

In order to facilitate semantic retrieval at multiple abstraction levels, annotations are not strictly assigned to semantic categories, but are arranged in an internal weighted representation to encode the hierarchical annotation information and to express the relations and similarity between the underlying images. Thus, using an existing annotation ontology and a set of specific rules, a space of annotations (*annotation space*) is built for the subsequent derivation of connections between images. Figure 4.6 visualizes a small example of the semantic network constructed for the annotations Γ_1 and Γ_2 presented in Figure 4.2.

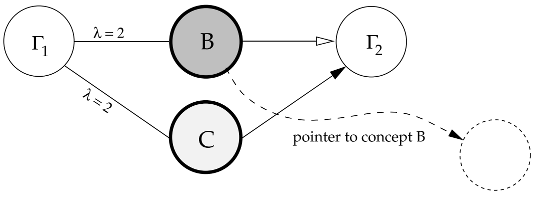


Figure 4.6: Representation of Specialization/Genaralization as a Multi-Graph

Formally, the network consists of nodes $V_\Gamma = \{\Gamma_1, \ldots, \Gamma_z\}$ which correspond to the image annotations $\Gamma_1, \ldots, \Gamma_z$ and a set of edges $\{e_1, \ldots, e_m\} \in E \subset V_\Gamma \times V_\Gamma$ connecting them. For each concept X two annotations have in common, their nodes are connected by an edge $e[\Gamma_1, \Gamma_2]^{[X]}$ which is labeled by the concept X. There is a distinction between two types of edges:

- **subsumption edge** e_{sub}: denotes the stronger specificity of the respective concept in the annotation. The arrow direction points to the more specific annotation.
- **expansion edge** e_{ext}: expands the annotation by a new concept which represents additional information derived from the annotation ontology.

Consequently, the stronger specificity of the concept B in the annotation Γ_2 is visualized by a subsumption edge (white arrowhead) $e_{sub}[B]$, because the concept 'building' (B) is more general than 'library' (L) according to the annotation ontology illustrated in Figure 4.2. By using the *expansion edge* $e_{ext}[C]$ (black arrowhead) the semantic annotation is expanded by a new concept. For example, the fact that the entity 'Duesseldorf' is connected with the concept 'city' (C) by the `is-inst` relation is used to derive this additional information.

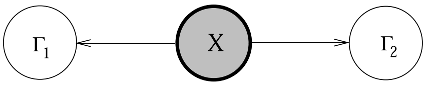


Figure 4.7: Syntactic (Synonymy) Relation between Annotation Terms

Syntactic relations, like synonymy, where the meaning of two or more terms is considered to be the same, are connected by their super-concept determined from the annotation ontology (see Figure 4.7). Example: *notebook* ← 'computer' → *laptop.* Abbreviated terms and their full forms are also treated as synonyms.

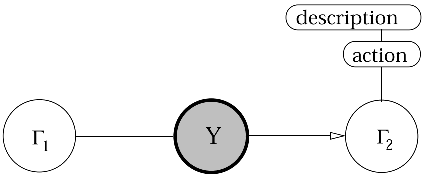


Figure 4.8: Representing Descriptive Features

Special features describing the image's content in more detail can be expressed by attributes, which are attached to the nodes in the annotation graph. An object annotated with keyword k_1 is characterized by additional descriptive attributes such as k_1.**color**:*orange* or expressing an action performed by the agent, like k_1.**action**:*eating* or k_1.**action**:*laughing.* The arrow indicates the direction of the relationship between nodes, in this case from the general to more specific node content.

In addition, the network (see Figure 4.6) is parameterized with the value $\lambda \in [0, \nu]$, denoting the *level* of the semantic relation between the annotations which is inferred from the ontology. For example, Γ_2 is extended by the concept 'city' (C) which is situated at a higher semantic level according to the hierarchy. The corresponding level is represented by the maximum distance between the individual keywords which are associated with the concept 'city' (in our example, this maximum distance is provided by the path (C) → (B) → (L)). Formally, the value of λ for an edge $e[\Gamma_a, \Gamma_b]^{[X]}$ is computed by the following formula:

$$\lambda(e[\Gamma_a, \Gamma_b]^{[X]}) = \max(\forall_{[C_i, C_j]^{[X]}} \text{dist}_{O_{d_i}}(C_i, C_j)), \tag{4.1}$$

where $[C_i, C_j]^{[X]}$ denotes a shortest path between concepts $C_i \in \Gamma_a$ and $C_j \in \Gamma_b$ via the node X and $\text{dist}_{O_{d_i}}(C_i, C_j)$ represents the distance between two concepts C_i and C_j in the annotation ontology O_{d_i}. Thus, the overall similarity of two annotations depends on the number of their connections and the levels λ between the used concepts.

Expressivity and quality of annotations play an important role for annotation systems. Therefore, two further measures are introduced in Table 4.2 to indicate the goodness of an annotation. The *specificity* σ is quantified to a positive real number

measure	
$\sigma(\Gamma_a) = \lvert e_{sub} \rvert \times \frac{1}{\mathcal{N}(\Gamma_a)}$	*preciseness*
$\epsilon(\Gamma_a) = \frac{1}{n} \sum\limits_{\Gamma_i \in N(\Gamma_a)} \text{dist}(I_a, I_i) \times \Lambda(\Gamma_a, \Gamma_i)$	*visual expressiveness*

Table 4.2: Measures for Annotation Quality

$\sigma(\Gamma) \in \mathbb{R}^+$ and is based on the fact, that annotations with a high number of subsumption edges describe more specifically the image contents and the more specifically described is an image the fewer images with this content will exist in the data collection. For example, if there are only two images with a particular annotation, then we can assume, that the annotation is very specific. Therefore, $\sigma(\Gamma)$ is computed by dividing the number of subsumption edges by the number of images attached with this annotation.

The second measure reflects the *visual expressiveness* ϵ of an annotation, specifying to what extent the used annotations have visual characteristics. The smaller this value is, the more discriminative power at feature level is provided by the annotation. This information is important for image annotation, especially for (semi-)automatic image annotation, since not all concepts are related to visual contents. This characteristics firstly depends on the number of keywords which have been assigned to image segments (with respect to low-level features). If this information is unavailable, it can be intuitively concluded that concepts described by annotations which are close to each other in the annotation space and whose images have similar visual characteristics have more discriminative properties than similar annotations specifying images with high discrepancy at feature level. The value of ϵ is therefore computed by the formula presented in Table 4.2, where

$$\Lambda(\Gamma_a, \Gamma_i) = \exp\,(-\frac{\|\Gamma_a - \Gamma_i\|^2}{2\delta^2})\,, \tag{4.2}$$

$\|\Gamma_a - \Gamma_i\|$ denotes the semantic distance between two annotations, dist$(\cdot)$ the images' distance at feature level, n the number of similar annotations within the neighborhood N, and δ the circumference of N. Thus, annotations in the neighborhood of Γ_a describing similar image contents are weighted according to their distance. In this case, if an annotation is close to Γ_a the 'penalty' of visual dissimilarity is high. In contrast, if the corresponding annotations are far away from the reference annotation Γ_a, the penalty will be decreased to zero, according to the *Gaussian* neighborhood function.

The advantages of the new representation are the following. First, implications about the semantic similarity of annotations can be determined by considering the incoming and outgoing edges in the multi-graph structure. In addition, relations like specialization can be discovered by considering the degree of the hierarchical distance. As demonstrated by the following Example 4.2, the resulting multi-graph structure is used to support semantic queries at different levels of abstraction.

Example 4.2 *Query Example.*
Let us suppose, that we have two annotations
$\Gamma_a = \{C_1 =$ 'Building', $k_1 =$ *Düsseldorf*$\}$ and
$\Gamma_b = \{k_1 =$ *Students*, k_1.**action:***learning*, $k_2 =$ *Düsseldorf*$\}$

When a user searches for *libraries* in Düsseldorf and specifies the query to the level $\lambda = 1$, the image annotated with Γ_b will appear in the result set, although this annotation does not contain directly the concept 'library'. Because of the existence of several other images liked to this concept, it follows from the graph structure a connection between 'library' and the activity *learning.*

4.4 Extending the Probabilistic Annotation by Multi-level Annotations

According to the probabilistic annotation approach, where keywords' relevance or importance for the characterization of an image is determined by the hypothesis that similar images may share the same keywords, the set of keywords for annotating I_q is determined by the following three steps:

i. Calculating the k most similar images $I_1, \ldots, I_k$ based on their low-level features,

ii. Statistically identifying frequent annotations associated with the k images, and

iii. Extending the results by taking into account the multi-level properties of annotations.

These three steps are illustrated in Figure 4.9. The set of images $I_1, \dots, I_k$, which is similar to a target image I_q is computed by applying the k-Nearest Neighbors algorithm (kNN). The detected images satisfy the criterion $sim(I_q, I_x) < \varepsilon$, where $sim(\cdot)$ $(0 \leq sim(\cdot) \leq 1)$ is the distance metrics computing the dissimilarity between two low-level feature vectors. The most suited annotations for the image I_q can be simply determined based on the annotations of its similar images. Assuming that we have an underlying probability distribution $P(\cdot|I_x)$ for each image $I_x \in \mathbf{D}$, which can be thought as a vector that contains the low-level features of the image, as well as all keywords $\{k_1^q, k_2^q, \dots, k_p^q\}$ that appear in the annotation of I_q. Due to the *Probabilistic Model* the probability $P(k_j|I_q)$ that a keyword k_j is suited for the annotation of the image I_q is defined as [CC03]:

$$P(k_j|I_q) = \frac{w_j}{\sum\limits_{j'=1\dots m} w_{j'}}, \tag{4.3}$$

where w_j is the weight of keyword k_j, which is computed as following:

$$w_j = \sum_{\forall i} sim(I_q, I_i) \times \beta_{ij}, \tag{4.4}$$

where $sim(I_q, I_i)$ represents the similarity value between the images I_q and I_i and β_{ij} $(0 \leq \beta_{ij} \leq 1)$ defines the importance of the keyword k_j to the image I_i. This importance can be estimated by a modification of the *tf.idf* weighting, namely by the frequency of the word k_j in annotations of similar images multiplied by the inverse frequency of this keyword in other annotations. Nevertheless, this approach does not consider the fact that keywords are related to each other and are linked to concepts, giving the keywords *meaning* at a higher semantic level. In addition, some keywords describing emotions or actions (e.g. 'driving') are difficult to be associated with visual features.

To alleviate the deficiencies mentioned previously, another third step is needed to enhance the results of the automatic annotation. Now, the existing connections between keywords and concepts are considered and the annotation space is used to evaluate the relations between annotations. For example, by examining the relations between several annotations containing a given keyword k_j, its importance for the description of a concept can be inferred. Thus, the probability that a given keyword k_j in a given context l, abbreviated by $P([k_j, l]|I_q)$, will be accurate for the annotation of the image I_q is defined by Formula 4.5. In this case, the context l is defined by a set of concepts. Later in Chapter 5, the context will be supplemented by an optional set

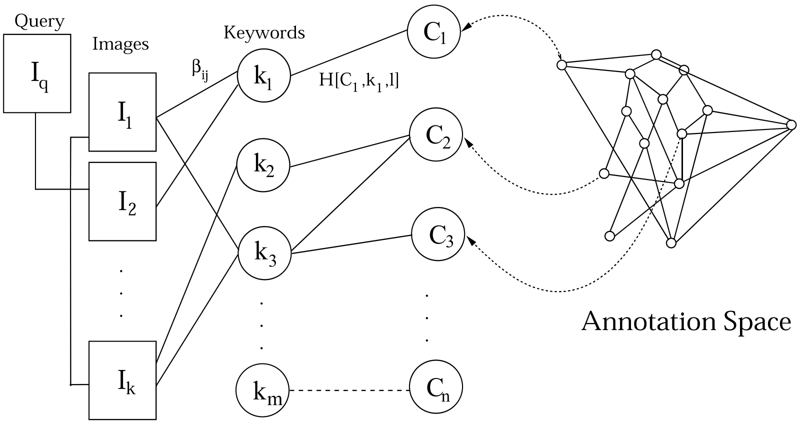


Figure 4.9: Linking Keywords to the Annotation Space

of negative constraints, which comprises keywords or concepts to be excluded.

$$P([k_j,l]|I_q) = \frac{w_j^{new}}{\sum\limits_{j'=1...m} w_{j'}^{new}}, \tag{4.5}$$

where w_j^{new} is computed as following:

$$w_j^{new} = \sum_{i=1...n} H[C_i,k_j,l] \times (\sum_{j'=1...m} w_{j'} \times H[C_i,k_{j'},l]), \tag{4.6}$$

and represents the weighted sum of the concept weights. The w_j's are the weights computed in Equation 4.4, and $H[C_i,k_j,l] \in [0..1]$ denotes the relevance of the keyword k_j for the description of the concept C_i depending on a given context l. There are many possibilities to determine this relevance. First, it can be inferred from our graph representation by considering the existing relations between annotations, like specialization, syntactic relations, or the quality measures. For example, to infer $H[C_i,k_j,l]$ from our representation, the number occurrences of the keyword k_j in the context l is determined as the number of annotations connected by an edge labeled with l.
To sum up, if a keyword k_j had a low probability in the classical approach, it can be increased by the fact, that it is frequently used in annotations of similar images or by frequently having a relation to other frequent annotations within the same concept.

4.5 Related Work

Image annotation still remains indispensable in most retrieval systems, although it is still associated with a relatively high degree of uncertainty. Bruce and Hillmann list seven metadata quality criteria in their work [BH04]: *'completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness,* *and accessibility'*.

In order to near the required characteristics, several standards have been proposed and used in the literature for the representation of multimedia document descriptions and their semantic interpretation, for example the *Dublin Core* [Cam02], *MPEG-7* [SS02, MSKP02], or *MPEG-21* [BGP03]. The main focus of the above standards is to provide a set of predefined categories and types of metadata, which are used for the description of multimedia contents subsequently allowing interoperable searching, indexing, filtering, and browsing of audio-visual content. The introduction of the MPEG-7 standard has been an important evolution in modeling and representing the audiovisual content. MPEG-7 uses several XML-based *Descriptors* (Ds), that are used to describe the various features of multimedia contents and *Description Schemes* (DSs) providing pre-defined structures for descriptors and their relationships. However, the usage of XML in combination with MPEG-7 (e.g. in [LKB+02]) does not provide any reasoning functions allowing the deduction of facts from multimedia descriptions. Although this approach can be perfectly applied to the structural description of the data and to metadata, it is rather unsuitable for the extended semantic description of their contents. The reason for this lies in the static descriptions which do not provide a formal semantics and cannot be processed by inference making facilities. In addition, probable inconsistencies, ambiguities, or duplication among the MPEG-7 descriptor schemes and descriptors cannot be discovered, because MPEG-7 does not provide the solution to model the whole annotation knowledge. A further disadvantage of such XML descriptions are however, that they can only be correctly retrieved by a adapted query language. For example in [KKK03, TC02], XQuery is used as the retrieval language retrieval of XML-based documents. However, users are demanded to have an advanced knowledge of the MPEG-7 details in order to express a precise query, and queries are easily getting complex. Analogously to data retrieval (see Section 2.1.2), such query languages are rather suitable for structured data since the returned query results satisfy clearly defined conditions. Instead, we would need methods which take into account different possible users' interpretation of the content and would be aware of the high-level requirements of the users.

Our work differs from the mentioned approaches through its focus on users' subjectivity which implicates special requirements, such as the detection of equal content descriptions at different abstraction levels. Our approach analyzes and evaluates the annotations given by different users and returns useful information about the underlying data collection, that cannot be found in the annotation ontology. Through this preprocessing of semantic information, the mappings of the low-level features into semantic concepts can be improved, leading to an increase of precision in image retrieval and semi-automatic annotation methodologies.

4.6 Summary and Future Work

In this chapter we have demonstrated existing problems in the field of image retrieval supported by semantic annotation. In the main part we have introduced the Multi-level Annotation Component which analyzes and evaluates the assigned multi-level annotations at both feature level and semantic level. The resulting semantic information is transformed into a multi-graph representation, which encodes the complex structure of hierarchical semantic relations and discovers similarities between differently annotated images. The information derived from this representation can be easily utilized to supplement existing annotation models (e.g. the probabilistic model) and to allow a context-based similarity evaluation between keywords and different annotations. Another promising aim is to automatically detect annotation inconsistencies within image collections or to use our approach for the creation of correctly annotated image data corpora (as training data) which are the basis for the evaluation of annotation-based retrieval systems.

5

Extracting Contextual Information from Multiuser Retrieval Systems

Although the most existing keyword-based systems are expanded by conceptual knowledge (e.g. ontologies) in order to model the topics in which the user is interested in, there still remain some unresolved problems, like existing differences in interpretation of image contents or inconsistencies in keyword assignments among different users. In our approach, multiple sources of information, which are modeled as different user profiles and annotation ontologies, are brought together in order to extract contextual information, and consequently to attenuate users' subjectivity occurring during querying and content describing. At the same time, this subjectivity serves as an instrument for semantic query expansion preventing the retrieval to fail in case of different perspectives on image collections. Finally, we evaluate our context-based approach on a real data set of sports images and the query expansion approach on a test collection of news data. The experiments demonstrate considerable retrieval quality, already in the first search iteration, which makes an additional query refinement dispensable. The results can even be further improved by applying lexical analysis for strings and error elimination methods.

5.1 Motivation

The amount of image and textual data has increased in recent years, ranging from personal photo collections to professional news and documentary archives. This trend has brought a great demand for intuitive retrieval and browsing facilities supported by semantic data classification and methods for efficient information sharing among thousands of users. The access to image data is mostly realized by content-based image retrieval based on low-level image features which can barely express the users' information need. The reason for this problem is the semantic gap which forms a major challenge in image retrieval and is defined as the discrepancy between the (high-level) meaning that a user demands and the features that can be automatically extracted from the underlying data. In order to facilitate queries at semantic level, several approaches, like [CC03, WDS$^+$01, DBdFF02, TPCR04, VC05a], have been proposed which combine automated feature extraction approaches with concept-based or annotation-based techniques. Their main objective is to (semi-)automatically attach images with appropriate descriptions and thus support content-based retrieval by a keyword-based search. In addition, images can be seen as instances of a complex ontology allowing the specification of objects and actions depicted in images and their classification into one of the predefined categories (e.g. outdoor, cars, faces, etc.). Thus, this approach facilitates concept-based search instead of a keyword search and allows users to specialize or generalize queries with the help of a concept hierarchy.

However, in such systems operating with high-level knowledge, there appear another manifestations of the semantic gap, particularly when data is processed and annotated within multiuser systems. In order to make manually created image annotations useful for efficient retrieval, some disadvantages have to be eliminated. The first disruptive factor is the fact that users do not have the same background knowledge or conceptual view on the data collection, resulting in a complicated access to relevant information and query formulation. Therefore, there exists the need for methods which effectively manage the increasing annotation data and possess the ability to automatically discover differences in interpretation of image contents or inconsistencies in the keyword assignments among different annotators. For example, the incidence of different contexts is expressed if the same image is assigned to different topics by two different users. Another type of *context mismatch* between users might occur during the query, for example if users' preferences, linguistic differences or the usage of different abstraction levels for the search influence the formulation of the same information need. On the other side, the users' subjectivity can be sometimes useful for interpreting images. After a successful determination of correspondences between the system's terminology and the user's vocabulary, the subjectivity can be used for knowledge expansion and query modification according to the user profile. For the most part, the subjective context (e.g. keywords *'laughing'* and *'children'*) is accessible only by the annotation words assigned to images [VC06]. Thus, subjectivity may enhance semantic retrieval when there exist methods to understand and interpret the characteristics of the assigned annotations.

In this chapter we consider two different aspects [VSC08, VC08a]: First, multiple sources of information which are modeled as different user profiles and annotation ontologies are brought together in order to extract contextual information and attenuate users' subjectivity. The second issue is to provide an interface for querying the complex data without understanding the whole system's terminology and to prevent the retrieval process to fail in the case of different views on the collection. For this purpose, the information of users' annotations is used to find mappings between the system's ontology and the user's vocabulary and thus to infer additional query parameters for a query reformulation.

This chapter is organized as follows: In Section 5.2 we describe the formal preliminaries of the image annotation process, the procedure to develop ontologies, and the aims of concept-based retrieval. Section 5.3 introduces the semantic context and shows how the similarity between image annotations and the posed query is computed subject to a given context. Beside this, a strategy to unify the core annotation ontology and the user-dependent knowledge by finding corresponding concepts is presented and it is shown how to reformulate the query according to the discovered mappings. Afterwards, the functionalities of our complete retrieval/annotation system, including the GLENARVAN [VSC08] component are presented. GLENARVAN's main task is the storage and the management of the annotation data, and the execution of preprocessing steps needed for the determination of context-based similarity. Finally, a set of experiments on a real-world domain evaluating the effectiveness of our approach is performed. In a second evaluation (Section 5.4), we used news data which allows efficient derivation of 'annotations', and is proved to be a suitable for validating the presented query expansion method. Section 5.5 separates our work from other relevant related papers. Finally, we conclude our approach in Section 5.6 and give future research directions.

5.2 Annotation-based Retrieval

The assignment of *terms* or *keywords* to images for capturing their semantic contents and enriching them by additional information is known as *image annotation.* In order to combine the high-level tasks of *scene recognition* and *user interpretation* with traditional CBIR systems, the manual annotation is performed by users. Accordingly, the general image annotation process includes the following steps:

1. Analyzing the image contents in order to identify relevant objects or regions and their relations.

2. Determining a set of candidate keywords for the annotation of the image by using an application-specific lexicon.

3. Assigning a set of keywords to the image at different abstraction levels, for example by describing the recognized objects, their relations, and the overall classification of the scene.

5.2.1 Definitions

Formally, for concept-based image annotation we need a set of semantic concepts $\Phi = \{C_1, C_2, \ldots, C_n\}$ which are arranged by means of inheritance and abstraction in an ontology O_Φ. The subset relation $\subseteq_\Phi \subset \Phi \times \Phi$ is a partial order between concepts ($C_i \subseteq_\Phi C_j$, which denotes that C_i is a sub-concept of C_j). The set of concepts is not known apriori and is dynamically extended when a new concept is added. An annotation Γ of an image I is represented by a set of keywords from $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$. The subset $\mathcal{K}_d \subseteq \mathcal{K}$ is a set of l keywords or semantic labels $\{k_{d1}, k_{d1+1}, \ldots, k_{dl}\}$ which are used in an application domain $d \in \mathcal{D}$. Since the annotation Γ assigns the data into a semantic category, keywords represent a description for a *concept instance* at multiple abstraction levels.

For this work, we need additional definitions of *users' profiles*, *context* and the term *multi-context*. A profile $\mathcal{P}$ of a user u is modeled as a tuple

$$\mathcal{P}_u = (O_u, \mathcal{L}^u), \tag{5.1}$$

where O_u denotes user's ontology used for the annotation, classification, and retrieval from the data collection, and

$$\mathcal{L}^u = \{l_1^u, l_2^u, \ldots, l_s^u\} \tag{5.2}$$

represents a set of user-specific contexts. A certain user context $l^u(q)$ for a query q is defined by a set of concepts and an optional set of negative constraints, which comprises keywords or concepts to be excluded in a query. These constraints are selected by the user during the interaction with the system. Based on user behavior, a specific context in the user profile can be updated or a new context can be added. Such a user profile is utilized to provide the user with his/her own annotation ontology that is more consistent with his view of the world. In this regard, the term *multi-context* means that based on a multiuser retrieval system, data can be annotated and categorized subjectively by different users. The main task here is to find methods to resolve the aggravating multi-context, by incorporating the individual user profiles and by providing methods to understand the structure of the individual classification schemas for both retrieval and annotation purposes.

Example 5.1

Let us assume a scenario in which the user has started with a keyword query q using the search term *'jaguar'*. If this keyword can be matched against one of the existing concepts in the system's ontology, the system will present the potentially relevant nodes to the user. Now, if the user has selected the concept *'animals'* and selected *'cars'* as negative constraint, the context for his query will be set to $l^u(q) = \{\text{animals}, \neg\text{vehicle}, \neg\text{cars}\}$.

5.2.2 Modeling Ontologies for Annotation

Annotations should assign the image data to one or more of the predefined categories resulting in a semantic classification of the whole data collection. Ambiguous interpretations are avoided by using a lexicon-based knowledge (e.g. thesaurus) which serves as a source of semantic terms and their relations. A first frame for annotating the data collection is given by an ontology which describes abstract concepts and their interrelationships and thus provides an abstract view of the application domain. Semantic Web techniques [BLHL01a] provide a platform for defining class terminologies with well-defined semantics and a flexible data model for representing metadata descriptions. In our application, the annotation ontologies are modeled using RDF Schema [LS], which defines ontology classes in a hierarchical manner. In addition, the Resource Description Framework, RDF [BG00], can be used both for annotating image metadata and visual features according to the ontology. The ontology together with the metadata forms an RDF graph, a knowledge base, which facilitates new semantic retrieval based on inference. In our study, we only used the RDF Schema approach to describe ontological models of the concepts involved in the image repository and for describing image contents. For example, an image depicting a football player named *Max Smith*, can be described in the following way:

```
<rdf:RDF
  xmlns:rdf="http://www.w3org/1999/02/22-rdf-syntax-ns#"
  xmlns:base="http://example.org/thinks#"
  xmlns:ex="http://example.org/schemas/sport#">
 <rdf:Description rdf:about="http://example.org/thinks#sample-image.jpg">
   <ex:hasPlace>Munich</ex:hasPlace>
   <ex:hasAction>scoring a goal</ex:hasAction>
   <ex:hasObject>ball, pitsch, spectactors </ex:hasObject>
   <ex:hasPerson>Max Smith</ex:hasPerson>
   <rdf:type rdf:resource="http://example.org/schemas/sport#munich"/>
 </rdf:Description>
...
</rdf:RDF>
```

Figure 5.1: Example of an RDF Annotation

Figure 5.2 represents an example of an *annotation ontology,* which is alternatively provided by the user or is interactively created during the retrieval. That means that the system initially provides a general ontology (*core ontology*) that includes only some fundamental concepts, and enables users to expand it according to their individual interests. The image collection itself in combination with a general ontology forms the basis for deriving a domain-specific ontology which covers a set of well-defined concepts. This ontology (subsequently named *annotation ontology*) is then used to annotate the existing image collection. Hence, the domain-specific ontology is mostly restricted only to concepts which are needed for classifying the image collection. By carefully choosing the number of concepts, a well arranged guide for a concept-based browsing in content-based systems can be achieved, and the complexity of similarity computation at semantic level can be minimized. As summary, the usage of annotation ontologies has the main objectives:

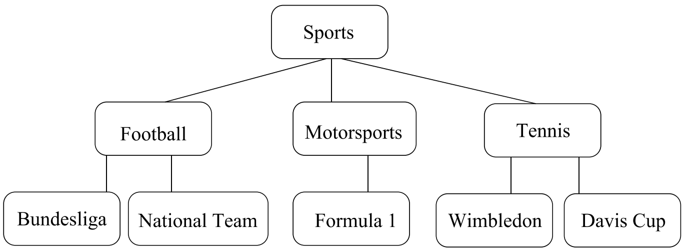


Figure 5.2: Concept Hierarchy as a Classification Schema for the Domain *'Sports'*

Concept-based Search. The ontology with its concepts and relations can be used to discover hidden semantic relations between a selected image and other images in the repository. Previously, such images would not be necessarily included in the answer set of the query.

Classification. Annotations assign images to one or more of the predefined categories resulting in a semantic grouping of the underlying data collection and thus providing a classification system for the organization of the data.

Keyword Finding. Keywords linked to ontology concepts provide an extended description of the data. Furthermore, from the assigned keywords we can generally infer relevance probabilities which are afterwards required for the (semi–) automatic annotation of unknown images.

For visualizing the concepts, there are many ways to represent concepts and their relationships in an ontology, however, in our developed system we used the hierarchical tree structure representation as a simple and compact visualization. Figure 5.3 shows an excerpt of a core ontology, coarsely dividing the application domains. The ontology

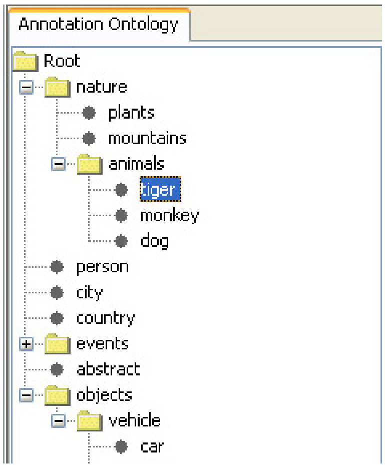


Figure 5.3: 'Core' Ontology

tree is presented to the user during the image retrieval and helps to track the semantic nature of the data collection. In RDF, each concept is defined by its class name and class label, and each class name is unique while a class label can be shared by several classes. Therefore, only the class labels are displayed in the ontology tree. Each subordinate connection between two concepts denotes a sub/super concept relationship. The core ontology provides the user a first basis for 'fine-grained' classification according to his personal preferences.

5.3 The Problem of Multi-Context

In this section, we address the problem emerging during annotation-based retrieval based on the usage of *one* global ontology within a multiuser environment. The annotation-based approach seems to be problematic since it assumes that users have the same background knowledge and operate in the same contexts. Thus, the problem of handling *multi-context* [McD97] can be divided into two parts:

a) How can we model the contextual knowledge and its similarity?

b) How can we improve the retrieval performance by using users' subjectivity and the resulting mappings between users' conceptualization schemas to expand query parameters?

Application Examples

A. Searching. For example, let us assume that two users U_1 and U_2 are searching for photographs of the city of London using the search string $s =$*'London'*. The several abstraction levels appearing in image collections are simulated by two images, which are already annotated and stored in a database. The first one displays the *Tower Bridge* (annotation Γ_1, assigned to ontology class *CITY.Buildings*), and the other one presents the *Coat of Arms* (annotation Γ_2, class *CITY*) of London. Now, if we compute the respective context-based distances between the used search string s and the two images in the different contexts $l_1 =$ *building* and $l_2 =$ *city*, the distance between s and Γ_1 should be smaller than the distance to Γ_2 when the user's search is restricted to the context of *building* and vice versa.

B. Annotation Objectives. Let us assume, that two users u_1 and u_2 want to annotate an image for sharing. The intention of user u_1 is to share the data with other members of a community. Thus the annotation is performed according to a shared ontology that is agreed between the members to ensure that the annotation is consistent and the image can be efficiently retrieved. User u_2 wants to share his image with friends and attaches free-text annotation to express emotions and memories. In this case, no formal classification is done. For that purpose, the images of user u_2 can only be retrieved in the future by a standard keyword search.

5.3.1 Contextual Similarity

In order to model the contextual knowledge of different users, we need some definitions of contextual similarity. The intention of contextual similarity is to model the different contexts in which an image may appear in and incorporate them into relevance computation. The context is inferred from the structure of the underlying ontology and from the corresponding data already classified (e.g. images assigned to classes). If we consider the context l, we can define a contextual query q^c as a tuple which has the form:

$$q^c = (s, \mathcal{P}_u)\,, \tag{5.3}$$

where s denotes a query string consisting of a set of keywords or concepts and $\mathcal{P}_u$ represents the profile of a user u, including a set of user-specific contexts. For the ranking of the results, a similarity value $f \in [0, 1]$ is computed for each considered image $I_1 \ldots I_N$. This value is determined by the function f:

$$f(q^c, \Gamma_I) = f_1(s, \Gamma_I) \times f_2(l_i^u \in \mathcal{P}_u, \Gamma_I). \tag{5.4}$$

Γ_I is the annotation of the image, which is compared with the query. The both functions f_1 and f_2 return the values $\nu_1, \nu_2 \in [0, 1]$. The first function f_1 returns the lexical similarity to the query string, the second the similarity according to an arbitrary context l_i^u included in the user's profile $\mathcal{P}_u$. By the unweighted multiplication of the two values, it can be ensured that only images are returned as results, which share a high similarity in both criteria. The two values are computed using the following similarity paradigms:

Lexical similarity (function f_1): It measures the degree to which two words are similar. In first step, to guarantee a fault-tolerant search, an elementary string comparison (e.g. using Levenstein distance) is done. For a further determination of similarity, a lexical database that organizes nouns and their relationships (synonyms, homonyms, hyponyms, and hyperonyms) is considered.

Contextual similarity (function f_2): Since in the most cases the *meaning* of a piece of data cannot be expressed by only one concept, the annotation ontology is used to determine the *context-based* similarity between annotations by examining the contexts in which keywords appear in. In our system, we distinguish between three types of context computation:

1.) In the simplest case, the user chooses a concept from the given ontology as context. Then, the similarity between the search string s and the annotated images in the database is determined by considering their weighted relations to this context. Afterwards, this context is inserted into the user's profile.

2.) The other case is when the context cannot directly be read off from the ontology, which means that the user's declaration of context does not occur as a concept in the annotation ontology.

3.) In the third case, the context computation is performed in consideration of several ontologies (created by other users). In order to solve the problem of inconsistencies among different users, methods for the incorporation of multiple ontologies have to be provided. The three approaches are introduced in the following paragraphs.

Computation of Contextual Similarity

Considering the excerpt of an annotation ontology presented in Figure 5.4 which is transformed into a directed weighted graph, relations like specialization or generalization can be variably weighted in order to model the degree of similarity and thus to allow a probability weighting with a particular uncertainty. For example, if a user is looking for images depicting the player *Max Smith* in context of *National Team*, we can assume that his information need will predominantly include images assigned to the class `'National Team'` depicting this player. However, it is most likely that images assigned to class `'World Cup 2006'` will in some degree fulfill the query. Depending on the application field, different weights can be defined in order to model generalization (bottom-up) and specialization (top-down) relations. For example, if we set the generalization weight to 0.9 and the specialization weight to 0.5, the computation of the context-based distance between a search string and a given context is performed as

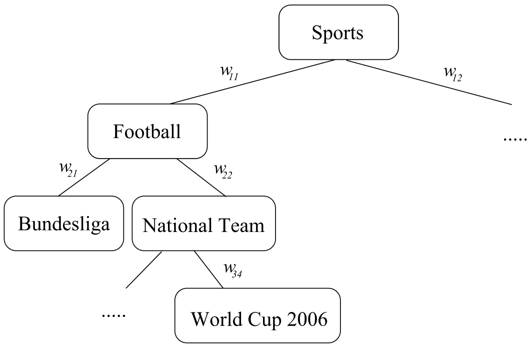


Figure 5.4: Weighted Relations between Concepts

follows: If the user is looking for the player *Max Smith* in context of *National Team*, relevant images which are assigned to the concept *Football* are scored with the value $\nu_2 = 0.9$ and images assigned to *Sports* with $\nu_2 = 0.9 \times 0.9 = 0.81$. Thus, the similarity is derived as the minimal weighted path length from the considered image to the concept representing the requested context.

Context Determination by Relevance Estimation

In case when the user's declaration of context is not selectable from annotation ontologies, a heuristics is applied to determine the most likely concept which could be applicable as starting point for the computation of contextual similarity (as described in Section 4.4). By introducing the relevance $H[C_i, k_j]$, denoting the importance of a keyword k_j for the description of the concept C_i, we can sequentially analyze the occurrences of the keyword k_j in existing annotations, and then determine the context obtained the maximum relevance value for this keyword. The relevance values are estimated by a modification of the *tf.idf* [BYRN99] weighting, namely by the frequency of the word k_j in annotations of a given concept multiplied by the inverse frequency of this keyword in other annotations.

5.3.2 Discovering Mappings between different Annotation Ontologies

In our approach, image features are divided into visual (low level) features and semantic (high level) features. Beyond the determination of correspondences between semantic concepts and visual low-level features, a set of rules (*mappings*) is constructed to identify the system's concepts which correspond to the user-specific vocabulary (mappings from O_c to O_u). These mappings together with a user profile are subsequently used to adjust the query parameters. In the first instance this problem can be traced back to the *ontology mapping* [DMDH02] task, where the aim is to integrate data from disparate ontologies, which use different terminologies/classifications of concepts in their taxonomies. The second step comprises the modification of the query according to the inferred mappings. In summary, the following steps are needed to expand the query:

1. to find correspondences between labeled data in the database according to the core ontology O_c and the user's ontology O_u by determining the similarity/dissimilarity values between concepts, and

2. to re-formulate the user's query according to these mappings.

Examples.

- Query Q_{old}:$\{c_{11}$=sandy beach $\wedge$ c_2=sea$\}$ is transformed into Q_{new}:$\{c_1$=beach $\wedge$ (c_2=sea $\vee$ c_3=ocean)$\}$ using existing mappings $subclass(c_{11}, c_1)$ and $synonym(c_2, c_3)$.

- Query Q_{old}:$\{s_1$='jaguar'$\}$ is transformed into Q_{new}:$\{s_1$='jaguar', $l^u(q) = \{$animals, $\neg$vehicle, $\neg$cars$\}\}$ using a user profile $\mathcal{P}_u$.

In the first example, the result set is both expanded by images assigned to concepts c_1 and c_3. Concept c_{11} is not taken into account for retrieval, since this category is not known for the system. In the second example, the query is expanded by information from the user profile, by setting constraints over a query.

Query Expansion Algorithm

At initial point, the set $\mathcal{I}^{train} = \{I_1, \ldots, I_T\}$ $(T < N)$ of images (or documents) representing instances, are assigned to semantic categories (classes) according to the core ontology. The membership of a document to a class is stored in the binary matrix $\mathbf{B} \in \{b_{i,j}\}_{T \times n^c}$, where n^c is the number of classes of the core ontology O_c. In order to model knowledge about the relatedness of the users' vocabulary, the similarity values between keywords are represented by a symmetric matrix $\mathbf{A} = [a_{i,j}]_{m \times m}$, where the element $a_{i,j} \geq 0$ represents the similarity between two keywords k_i and k_j based on their categorization and lexical relations. Analogously to the core ontology, the user-specific ontology O_u for the annotation is classified into a set of concepts $\Phi^u = \{C^u_1, C^u_2, \ldots, C^u_{n^u}\}$ linked with the used vocabulary - represented by the set $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$ of keywords. This co-occurrence information is organized in a matrix $\mathbf{D} = [d_{i,j}]_{n^u \times m}$, where the element $d_{i,j} \geq 0$ captures the frequency of the usage of vocabulary k_j in the context or description of a concept C^u_i.

In order to find correspondences between O_c and O_u, the respective similarity between two concepts is computed using the *Jaccard Similarity Coefficient* [Rij79], which is defined by:

$$\mathrm{JcSim}(X,Y) = \frac{P(X,Y)}{P(X,Y) + P(X,\overline{Y}) + P(\overline{X},Y)}. \tag{5.5}$$

The joint probability distributions between any two concepts X and Y consist of the four probabilities $P(X,Y)$, $P(\overline{X},\overline{Y})$, $P(\overline{X},Y)$, and $P(X,\overline{Y})$. For example, $P(X,Y)$ denotes the probability that an ontology instance belongs to both classes X and Y. Since in most cases, these probabilities are not available, they are obtained by learning from the data. The idea here is to take the concept examples from the ontology O_c as input for building a classifier, and then perform a cross-classification of the respective concepts from O_u into concepts of O_c and vice versa. The obtained scores represent the probabilities for inter-concept similarity. Since mapping results highly depend on the text classification algorithm, an appropriate choice of example instances and representative concepts is essential to facilitate accurate mappings.

Based on these preliminary considerations, a *query expansion algorithm* has been developed, which is presented in Figure 5.5: Function `compute_similarity` takes as input two ontologies O_c and O_u, together with their data instances and returns the similarity matrix between them denoting for every pair of concepts $X \in O_c, Y \in O_u$ their joint probability distribution. In a second step, the user's query Q_{old} is transformed into a new representation Q_{new} and sent to the system performing the query. The new query consists of the old concepts extended by corresponding concepts from O_c determined by the function *find_concepts* and keywords determined by lexical analysis, e.g. stemming or adding synonyms (function *find_keywords*).
Depending on the retrieval task, some feature-based constraints might be added as a weighting function, for example if low-level image similarity should be considered. To strengthen the influence of image features during the comparison of concepts, keywords' importance `Relevance`(k) can be estimated by incorporating the hypothesis that similar images may share similar descriptions. A method for this computation has been presented in Section 4.4.

5.3.3 System Components and Evaluation

$\mathcal{IKONA}$ Retrieval/Annotation System (illustrated in Figure 5.6) stands for our system architecture which provides functionalities for the *semi-automatic* annotation of multimedia data. Hence, its main feature is the component for *data annotation* which is used to associate the data with descriptors from existing ontologies. Furthermore, by interactive feedback and the analysis of the logical structure (low-level features) of already annotated images, the membership of the data to a predefined category is proposed. The application fields of $\mathcal{IKONA}$ cover domains involved with huge heterogeneous image/multimedia collections, whose content may be attached with semantic

```
function compute_similarity($O_c$, $O_u$){
   matrix[][] SimMatrix;
   for each ($X_i \in O_c, Y_j \in O_u$) do
      SimMatrix[i][j] $\leftarrow$ JcSim($X_i, Y_j$);
return SimMatrix;
}

function query_reformulate($Q_{old}$, $\mathcal{P}_u$, simMatrix)
   array[] newConcepts;
   array[] newKeywords;
   query $Q_{new}$:= $Q_{old}$;
   for each (concept $C^u$ $\in Q_{old}$) do
      for each (keyword $k^u$ $\in C^u$) do
        $find\_concept$ $C^c$ where
              JcSim($C^c$, $C^u$) < $\delta$ or $C^c$ derivable from $\mathcal{P}_u$;
          newConcepts:= newConcepts $\cup\{C^c\}$;
        $find\_keywords$   $k^c \in C^c$
          where Relevance($k^c$) $\geq$ Relevance($k^u$);
          newKeywords:= newKeywords $\cup\{k^c\}$;
      $Q_{new}$:= $Q_{new} \cup \{k^u\} \cup$ newConcepts;
return $Q_{new}$;
}
```

Figure 5.5: Annotation Mapping and Query Reformulation

meaning to become understandable and interpretable for the machine. Our IR System is extended by the GLENARVAN component, which is responsible for all the functionalities described in this chapter, like context computation, ontology comparison, and query expansion. The main tasks of GLENARVAN are:

- As initialization, RDF models are loaded from the existing annotations (annotation models) and the annotation ontologies (ontology models) stored in the system. During the query, the search component takes the ontology models with the posed query and the users' annotation models as input to generate a result list of relevance values determined by the function f (introduced in Section 5.3.1).

- Within the function f, the return value of the function f_1 results from string comparison (lexical similarity) by semantic comparison (sematic similarity). The semantic similarity is determined by taking into consideration the application specific lexicon (SportsNET), which was designed for our system. This lexicon includes pairs of terms with the degree of their relationship (expressed as weights in the interval $[0, 1]$). An example for a synonym relation is the name of a sports club and its abbreviation or a player's name and his nickname.

- The context parameter provided by f_2 is determined by the OntologyRating component, which uses the algorithm described in Section 5.3.1.

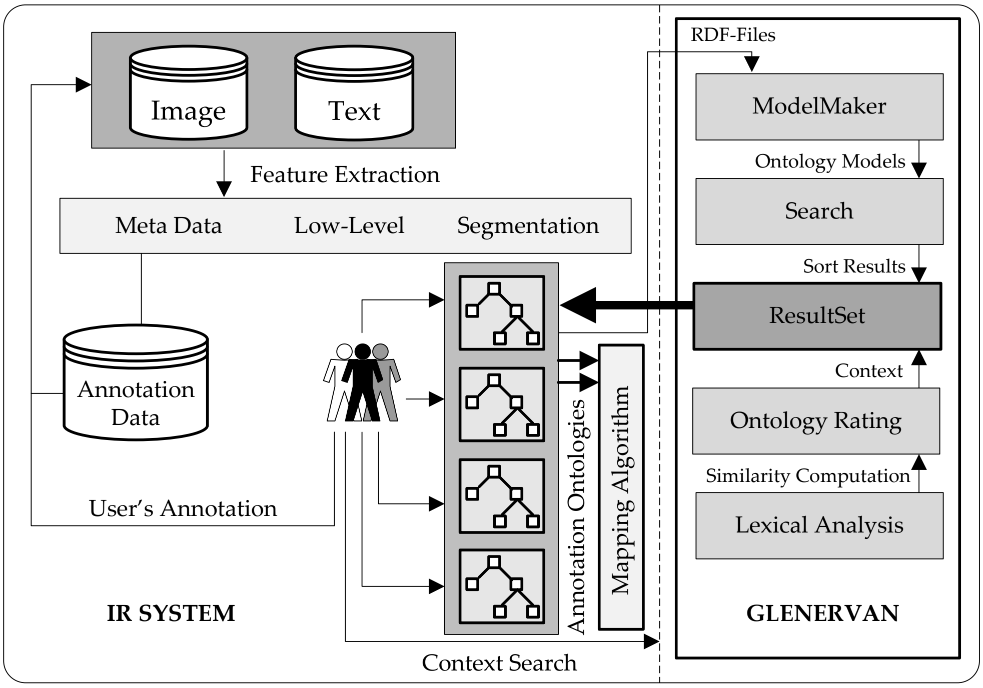


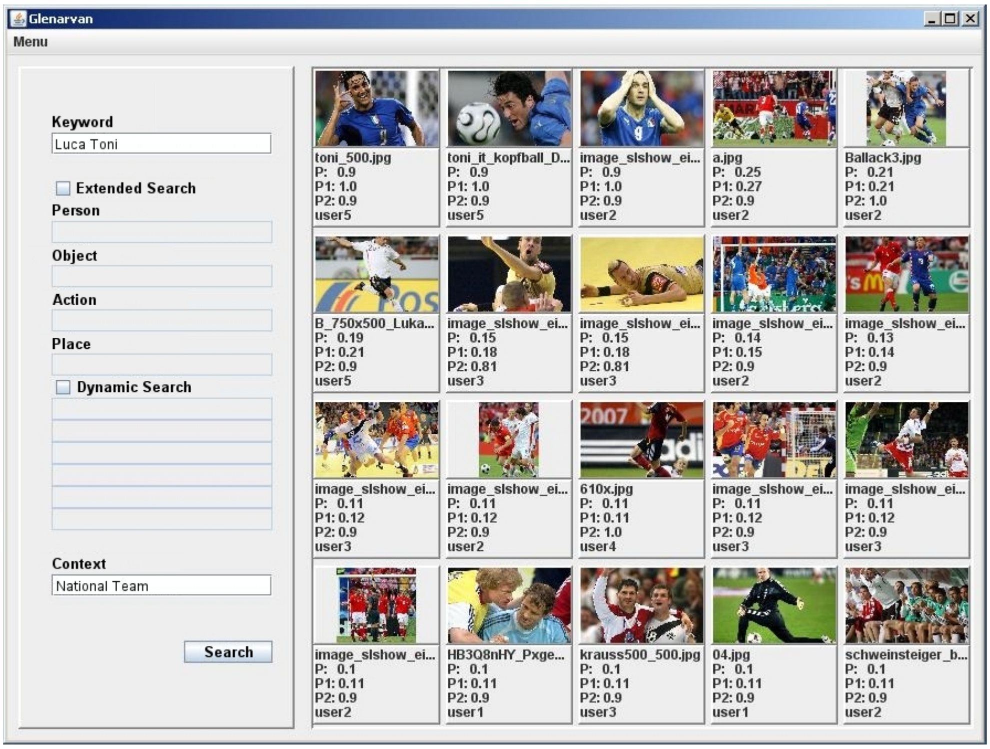
Figure 5.6: GLENARVAN System Architecture

Beside the determination of basic *meta data* (e.g. date, creator, or filename) from images, our system supports methods for extracting primitive (visual) characteristics of images. The set of *low-level* features currently includes color features, color moments, and texture characteristics. As graphical user interface, the system provides windows for interactively attaching images with semantic descriptions (partially taken from the *annotation data*). It also provides fields for a structured description, like person, object, and action properties. Furthermore, it visualizes the query results with the relevance values according to a given query. The graphical result visualization is presented in Figure 5.7. The system also stores different annotation ontologies, which can be subsequently adjusted to a 'fine-grained' classification according to the users' personal preferences.

Experimental Set Up

Due to the complexity of the problem, non-experimental methods for retrieval evaluation are barely suitable. In the following, the practical experience in handling with contextual queries will be described qualitatively. In addition, this experimental study will provide a detailed characterization of the result sets with regard to different types of queries in a specific application domain. For the evaluation, the following questions are examined:

A. How is the precision/recall of finding all relevant images according to a query?



Keyword: Luca Toni

Context: National Team

toni_500.jpg P: 0.9 P1: 1.0 P2: 0.9 user5	toni_it_kopfball_D... P: 0.9 P1: 1.0 P2: 0.9 user5	image_slshow_ei... P: 0.9 P1: 1.0 P2: 0.9 user2	a.jpg P: 0.25 P1: 0.27 P2: 0.9 user2	Ballack3.jpg P: 0.21 P1: 0.21 P2: 1.0 user2
B_750x500_Luka... P: 0.19 P1: 0.21 P2: 0.9 user5	image_slshow_ei... P: 0.15 P1: 0.18 P2: 0.81 user3	image_slshow_ei... P: 0.15 P1: 0.18 P2: 0.81 user3	image_slshow_ei... P: 0.14 P1: 0.15 P2: 0.9 user2	image_slshow_ei... P: 0.13 P1: 0.14 P2: 0.9 user2
image_slshow_ei... P: 0.11 P1: 0.12 P2: 0.9 user3	image_slshow_ei... P: 0.11 P1: 0.12 P2: 0.9 user2	610x.jpg P: 0.11 P1: 0.11 P2: 1.0 user4	image_slshow_ei... P: 0.11 P1: 0.12 P2: 0.9 user3	image_slshow_ei... P: 0.11 P1: 0.12 P2: 0.9 user3
image_slshow_ei... P: 0.1 P1: 0.11 P2: 0.9 user2	HB3Q8nHY_Pxge... P: 0.1 P1: 0.11 P2: 0.9 user1	krauss500_500.jpg P: 0.1 P1: 0.11 P2: 0.9 user3	04.jpg P: 0.1 P1: 0.11 P2: 0.9 user1	schweinsteiger_b... P: 0.1 P1: 0.11 P2: 0.9 user2

Figure 5.7: Graphical Search Interface of the GLENARVAN Component

B. To what extent does contextual knowledge and the defined semantic relations improve the accuracy of annotation-based retrieval systems?

To answer these questions, a labeled data set from a sport image collection (including categories like football, handball, motor sports, tennis etc.) has been provided by a domain specialist. In order to model a multiuser environment, each of the images has been assigned to one or more of the 7 ontologies, which differ from each other in the number of concepts, the structure and abstraction level. In order to produce noise, concept mismatches generated randomly by class label variation and slightly modified image annotations were included in the users' profiles. The 30 queries were subdivided into three types summarized in Table 5.1.

Type	#Q	Aspects
1	18	ambiguous, different contexts
2	3	orthographical errors
3	9	extended query, definition of relations

Table 5.1: Query Types

The three query types were designed taking several aspects into consideration. In the first instance, the tasks were created for the purpose to confront users with difficulties they are faced in a real-life retrieval with standard search engines. Type 1 demands a very specific issue, which is further specified by a context. Type 2 will face the user with orthographic errors which often occur during the annotation. This means that the system will need to look up the entered keywords to find appropriate correspondences in the annotation data. The set of images returned by query type 3 has to be found by considering additional constraints, like the definition of relations (e.g. *'x plays football'*). An extract of the posed queries is illustrated in Table 5.2.

Type	Query	Context, *Relations*
1	Accident Berlin ...	Formula 1 Sports ...
2	Rudi Völer ...	Sports ...
3	Oliver Kahn Diego ...	Football, *screaming* Bremen, *cheering* ...

Table 5.2: Query Examples

Results

The results of the experiments are illustrated in Figures 5.8 and 5.9 as bar diagrams, summarizing the precision and recall statistics of each of the query type. Each bar presents the average value after *one* search iteration. In addition, the average number of images (# Images) considered in the result set is presented. This number is directly controlled by the threshold τ, which is dynamically determined by the f_1 and f_2 values. For the evaluation we experimentally investigated

$$\tau = 0.5 \times f_T\,, \tag{5.6}$$

where f_T denotes the f-value of the top ranked image in the result set. Due to the inhomogeneity of the recall levels (number of considered images) in the individual queries, the classical precision versus recall curves were non-applicable for the evaluation. Each of the left bars (light gray) is obtained by the context-based queries without lexical knowledge, the right ones with using the dictionary.

In all the query types, a high precision and recall value could be achieved. The best precision was achieved by query type 2, which only used the automatic error elimination. Due to the string matching algorithm, errors in the search parameters could be efficiently corrected, and thus, the matching annotations could be determined. Considering queries of type 1, a high precision value (on average 87,10% and 97,89%) could be reached. The values could even be slightly increased by the SportsNet lexicon,

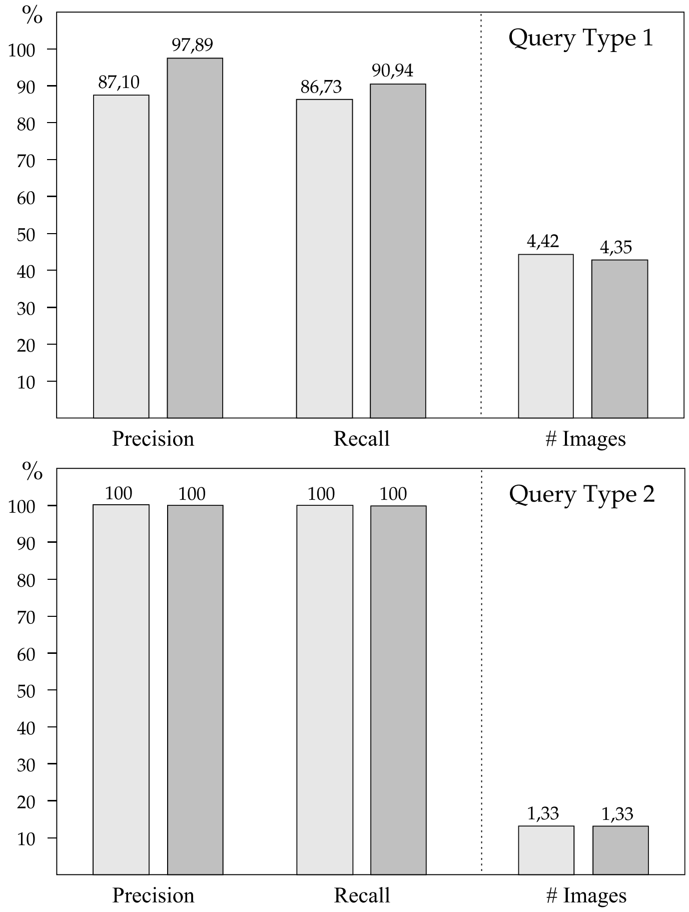


Figure 5.8: Average Precision and Recall Values for Queries 1 and 2

which provided relations and alternative keywords. With regard to recall, the behavior of extended queries (query type 3) was similar to the first, but in case of precision it performed worse (54,16% and 58,33%). The reason for this effect can be traced back to missing annotations and specializations (e.g. action=*screaming*) and the large number of relations to be considered, which resulted in an overloaded result set. Here, the probability of finding irrelevant images which have been incorrectly added into the result set is higher. The parameter τ provided a strong limitation of the number of images to be considered from the result set without having a negative effect on the precision. Thereby, it can be confirmed that the obtained result sets are very precise, resulting from the system's ability to transform a simple *'keyword+context'* query into a high selective query obviating ambiguous results. In addition, in the case when the context could not be directly determined (context did not occur as a concept in the annotation ontology), the analysis of existing annotations assigned to a particular topic helped to find the most likely semantic class to be considered as context.

In general, the results suggest that the usage of contextual information is helpful,

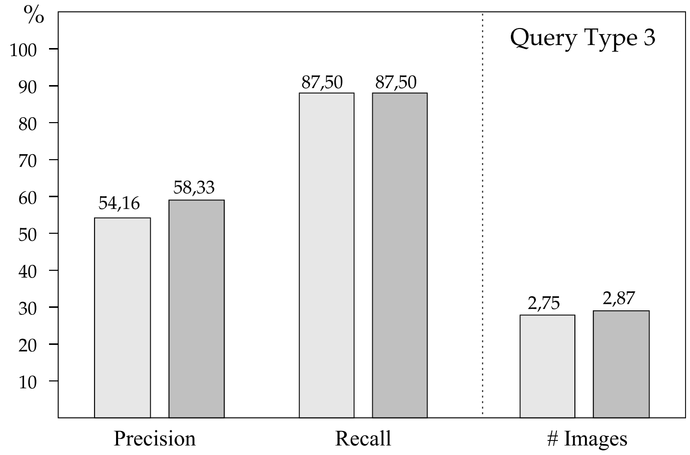


Figure 5.9: Average Precision and Recall Values for Query 3

particulary if the data collection is unknown and the system contains annotations created by multiple users with the help of different annotation ontologies. We have also noticed, that results below the relevance threshold τ, which have been incorrectly classified as non relevant (false negatives), have actually narrowly missed the result set. On the other hand, non relevant images found in the result set (false positives) seemed to have a semantic relationship to the demanded images – a fact which could be helpful for the user to get a general idea of the data collection and if necessary refine the query according to his new information need. As a summary, the property of vagueness resulting from the consideration of variably structured ontologies and the incorporation of users' subjectivity characterize our information retrieval system.

5.4 Query Adjustment by using User-dependent Annotation Preferences

The varying users' perception of image contents and the usage of different retrieval aspects make it necessary to develop methods for the unification and integration of different annotation schemes. In this section we put the main focus on the transformation of the subjective annotations assigned by different users into a unified knowledge base. The found correspondences between the already labeled data in the database and the user's ontology (and their vocabulary) are subsequently used to adjust a submitted query. This is done by the *query expansion algorithm*, which has been introduced in Section 5.3.2. The introduced method is separately evaluated on a large collection of news data including both images and the corresponding textual data. The experiments show that the reformulated queries significantly increase the retrieval quality, and thus prevent the retrieval process to fail in case of different sights on image collections.

Particularly when users are faced with a data repository whose content is unknown and has not been made completely semantically accessible, our method performs quite well.

5.4.1 Problem Description

In this section, we address the problem emerging during annotation-based retrieval based on the usage of *one* global ontology within a multiuser environment. The system's core ontology, which is used for generating suitable annotation patterns, results from a projection of the image feature space into a variable set of concepts and their qualitative characteristics from the knowledge base. This fixed ontology serves to obviate the inconsistency of keyword assignments among different indexers. It is also used to suggest users alternative terms for the description of image segments and helps them to better articulate and refine queries during image retrieval.

However, this approach assumes that users have the same background knowledge and interpretation ability. Since, this is not the case in real world applications, we model the subjectivity by different ontologies (in our experiment O_u^1 and O_u^2) created by a slightly modification of the system's core ontology. Test data is provided by manual annotation of randomly selected documents which are subsequently assigned as instances of a couple of concepts from O_u^1 and O_u^2. In the process of querying, this data is used both as an instrument for knowledge expansion and for finding correspondences between the system's terminology and user-specific conceptual views. Thus, the captured mappings between users' conceptualization schemas and the system are used to infer additional query parameters resulting in a better approximation of the user's information need. Figure 5.10 gives an overview of the experiment structure.

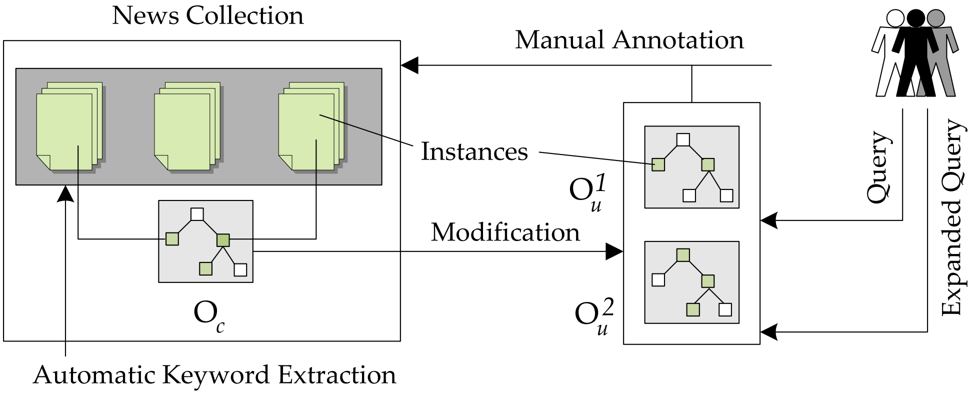


Figure 5.10: Experiment Structure

5.4.2 Experiments and Evaluation

In the experiments, the retrieval on a predefined data set was evaluated by comparing the retrieval *with* our query adaptation according to the personalized annotation ontology and *without* using the approach (keyword-based search using the vector space model). The two main criteria being considered for the evaluation of the effectiveness, are *precision* and *recall* [BYRN99], which mirror the accuracy of the system by measuring 1) the percentage of correct documents in the answer set and 2) the percentage of relevant documents found in the retrieval session. The values extracted from a set of queries are displayed as a curve of average precision at different recall levels (e.g. 10%, 20%, etc.). Since the aim of our approach is the query expansion according to users' subjectivity, the set of relevant documents for each topic and the relevance assessments of the obtained results was provided by the user himself – instead of using a fixed reference collection.

Experimental Set Up

As test data, a collection of 2.360 news articles was taken, which were crawled from news websites over the internet. The considered features comprised both image data as low-level features and the news abstracts as textual information. We defined a core annotation ontology for this data collection by partitioning the data into a set of important concepts and subconcepts (including e.g. *politics*, *science*, *countries*, *persons*, etc.) which are general enough to represent all data instances and corresponding subconcepts refining the taxonomy. Each concept of the system's ontology has been manually assigned a set of representative documents in order to extract the vocabulary for its description. This task has been done by using the *tf.idf* [BYRN99] weighting supplemented by a heuristic, which analyzes the structure and the formatting of the text, followed by a subsequent examination/clearing of the vocabulary.

Exp.	$\eta(O_c)$	$\eta(O_u)$	$\mathcal{K}(O_c)$	$\mathcal{K}(O_u)$
1 (O_c^1, O_u^1)	32	45	1–80	1–5
2 (O_c^1, O_u^2)	32	82	1–80	1–5
3 (O_c^2, O_u^1)	64	45	1–80	1–5
4 (O_c^2, O_u^2)	64	82	1–80	1–5

Table 5.3: Experiment Parameters

Table 5.3 lists the respective ontology parameters involved in the four experiment sessions. In reference to the two core ontologies O_c^1 and O_c^2, the users' ontologies O_u^1 and O_u^2 have been created according to the mentioned specifications by the manual definition of a modified concept taxonomy enriched by the respective keywords from the test data. $\eta(O_c)$ denotes the number of concepts (and subconcepts) in the core ontology and $\mathcal{K}(O_c)$ presents the average number of keywords determined for the description of a concept instance of O_c. Since the keywords have been extracted automatically from the news abstracts, their number was rather high (1–80 keywords). In contrast to this quantity, document examples associated to the users' ontology have been annotated with 1-5 keywords. The contents of the ontologies O_u and O_c have been chosen in that manner, that only a small percentage of *direct* concept overlap (see Table 5.4) would be provided for solving a given task. In order to produce noise, concept mismatches have been generated randomly by class label variation or by the movement of a sibling to a different parent. At instance level, a slightly modified vocabulary has been included in the users' ontologies.

Retrieval Tasks

The retrieval tasks `T1`–`T3` differed from each other in the abstraction level of the information need and were formulated as follows:

`T1`: Find information (image and text) about *Chancellors of Germany.*

`T2`: Find scientific articles (image and text) about *History of Earth and Evolution.*

`T3`: Find information (image and text) about *New York's Schools for Learning English.*

The three retrieval tasks have been designed under consideration of several aspects. In the first instance, the tasks have been created for the purpose to confront users with difficulties they are faced in a real-life retrieval with standard search engines. Task `T1` demanded a very general concept, which is not further specified and should return all chancellors of Germany. Task `T2` has faced the user with the *vocabulary problem*, which means that it is barely possible to find appropriate search terms for this information need. Here, the users had to rely on the lexical relations between keywords or the corresponding concept linked with a few relevant documents. The set of documents that should be found in retrieval task `T3` had a very specific character. The demanded news had to determine the names of particular schools and their location. An additional constraint here is was *learning English* activity.

Overlap	Task 1	Task 2	Task 3
O_c/O_u	32,4%	32,5%	17,0%
O_c/O_u	41,6%	21,6%	22,7%

Table 5.4: Average Concept Overlap in %

Results

The results of the experiments (tasks `T1`–`T3`) are illustrated in Figures 5.11 and 5.12 as average precision versus recall curves for both the classical and our approach (two curves for O_u^1 and O_u^2). In the first task, both approaches resulted in a high recall value. The best precision was achieved by query reformulation using the 'user ontology 1'. Due to the strong systematization of the concept 'politics' in both ontologies and a high occurrence of representative words in documents assigned to this topic, an appropriate mapping between the query words and the corresponding semantic classes could be easily found. The system's behavior in the second task was similar to the first, but in case of the smaller 'user ontology 1' our method performed worse than the classical approach. The reason for this effect is traced back to the coarse-grained user's concept set with only 45 concepts. The character of the defined information need was very professional but general, which could only be satisfied by a small number of documents. Consequently, the query reformulation was impaired by the fact that not enough representative documents for this topic were available in the data collection. The third task showed the limitations of our approach. The result set of this specific information need could not outperform the classical IR approach in precision, because of the missing representative documents for the concept school and the ambiguous class affiliation of words occurring in the representative documents.

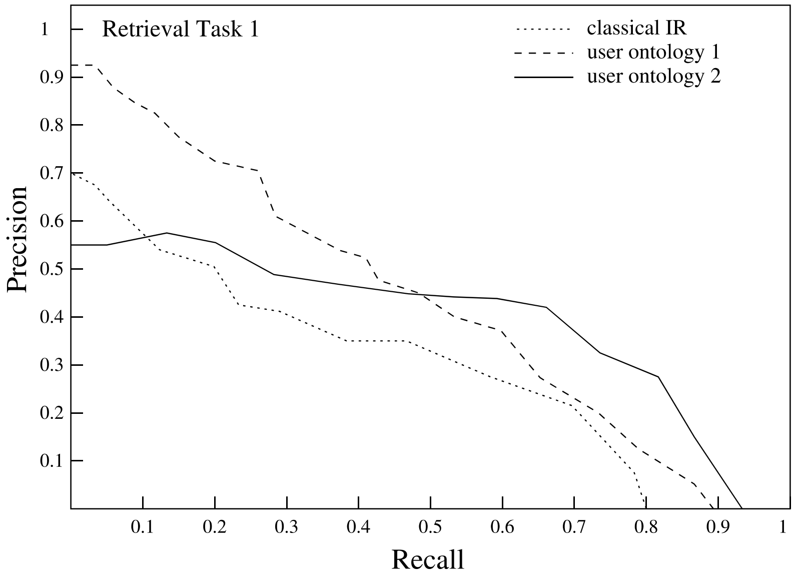


Figure 5.11: Evaluation of Retrieval Task T1

In summary, the results of our experiments show a substantial improvement of retrieval accuracy by on average 12,4% in the recall values. A slightly improvement of the precision by 6,2% could only be observed in task T1 and T2, indicating an adequate functionality in cases when enough data instances are available to ensure correct concept mappings. Generally speaking, the results suggest that we can efficiently incorporate personalized annotation ontologies to enhance the retrieval results.

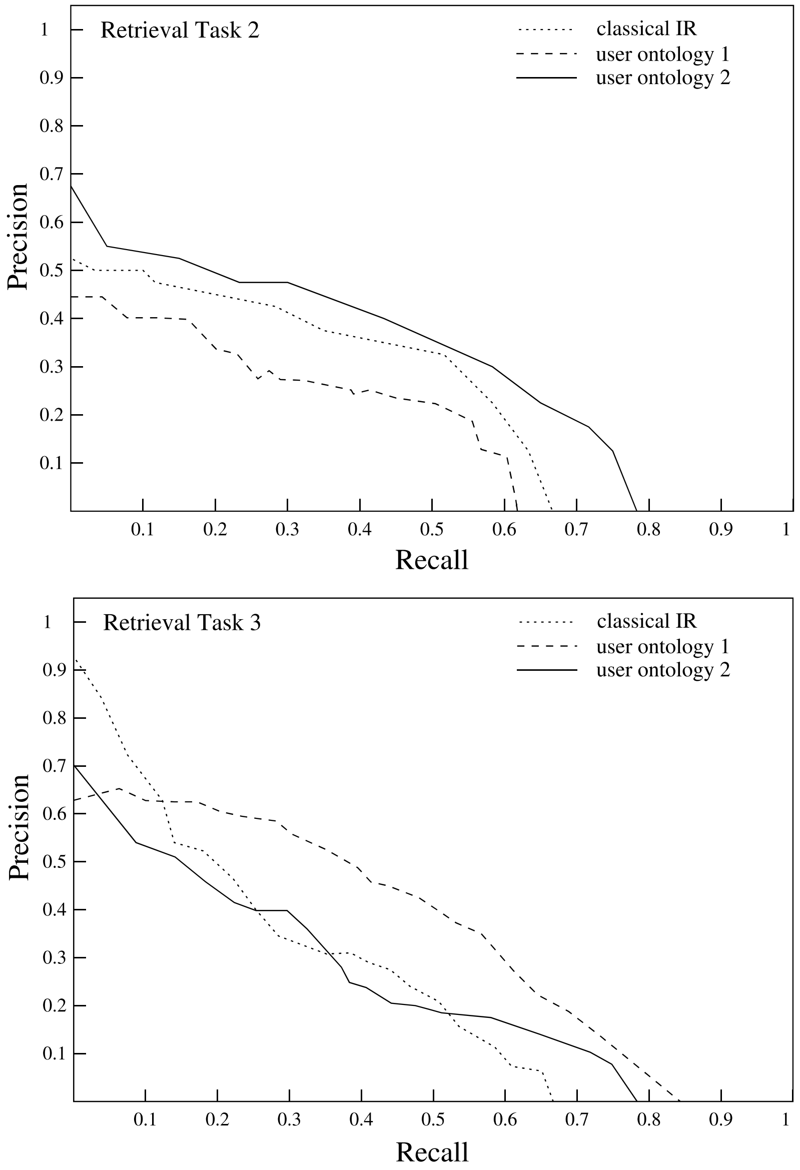


Figure 5.12: Evaluation of Retrieval Tasks `T2` and `T3`

5.5 Related Work

Due to the increasing usage of image sharing and retrieval systems and the rapid expansion of the world wide web, efficient information access in form of querying and browsing becomes increasingly essential. One of the key factors for an accurate information access is the *user context*. Hence, systems which know who is asking for information and for what purpose are in demand for providing the most appropriate answer to the user's information need. As characterized in [AAB$^+$03], interactions with web search engines could be characterized as *'one size fits all'*. This means that all users' requests are treated as static queries without any representation of user preferences, search context, or the task context. In James Alann's report [AAB$^+$03], contextual retrieval is defined as the task of combining search technologies and knowledge about query and user context into a single framework.

In recent research work, several enhancements of the pure querying on indexed data have been proposed. For example, methods for estimating the probability of document relevance to user queries, or determining weights for search terms have been studied in [RJ88, Wil92]. Ponte and Croft introduced in [PC98] their language model, where each document is represented by a document language model and each query is treated as a sample of text from the language model as well. The document result set is ranked according to the probability that the document language model could generate the query text. Relevance feedback approaches have presented the first attempt to incorporate users' interaction with the retrieval system. The question of to what degree relevance information can effectively be used by a relevance feedback process has been extensively studied in [BSA94]. A personalized query is constructed by re-weighting of the query terms based on some explicit or implicit feedback from the user [HR01]. In [WLWK06], the contexts of query terms inside a document have been additionally considered for the feedback in order to explore term co-occurrence relationships. Other approaches modify the initial query using words from top-ranked or as relevant identified documents. For example, in the mentioned language model [PC98], some additional words are added to the query based on the log ratio of the occurrence probability in the set of relevant documents to the probability in the whole collection. Another form of query expansion is done by lexical analysis, e.g. by including synonyms or closely related words into the query [CFPS02] or by resolving lexical ambiguity [KC92]. Nevertheless, synonym-based query expansion could be considered as a primitive form of applying domain knowledge. Although all these approaches enhance the retrieval quality to a certain degree, they are not satisfactory for disambiguate the sense of the user's query, defining query contexts and user models, which are central to personalization.

User's interests in web-based information access have been explored in several research work. For example, [BGG+99] introduced an agent for the exploration and (unsupervised) categorization of documents from the web based on a user profile. Lieberman presented in [Lie97] an autonomous interface agent that makes real-time suggestions for web pages that a user might be interested in and manipulates objects in the displayed interface, based on input implicitly collected from the user. Budzik [BBFH02] presented a system which can provide users with relevant resources in the context of their current work and thus help users with similar goals and interests to communicate both synchronously and asynchronously. The aspect of *annotation sharing* has been previously examined in [KK01]. The proposed annotation system is based on a general-purpose open RDF infrastructure, where annotations are modeled as a class of metadata and are viewed as statements about web documents assigned by users.

In [CLC06], Chakravarthy et al. presented AKTiveMedia, a user centric system for multimedia documents which allows users to annotate textual, image or multimedia documents in a collaborative way, sharing their experience with other members of the community. Language technologies are adopted to provide a context specific suggestion mechanism: for example when a user annotates a region of an image as *'part of an engine'*, the system suggests all the possible parts which are present in the ontology or in other user annotations. Appan et al. [ASSB05] investigate collaborative annotation systems for a network of users which has the aim of providing personalized recommendations which are inferred by a common sense inference toolkit.

These approaches, while the extracting user preferences, taking into account the users' behavior, and implementing recommendation methods based on inference, do not consider the modeled users' knowledge that can be used as additional source for the determination and disambiguating the context. Our approach combine the critical elements that make up a personalized retrieval system, by including the users' knowledge about the domain being investigated (in form of ontologies), the query expansion which can be seen as a short-term information need, and the captured user profiles which present the long-term interests of the user.

5.6 Summary and Conclusion

In this chapter we have presented an approach for supporting classical IR systems by modeling multiuser knowledge and profiles. The results suggest that we can efficiently incorporate contextual information modeled by ontologies to enhance the retrieval results. Thus, the retrieval quality can significantly be improved and a reduction of retrieval time can be achieved. The presented approach also facilitates the user to search through his own subjective view of semantic concepts, but concurrently utilizes other existing models for inferring additional query parameters. Furthermore, our approach can also be applied to analyze the users' annotation behavior. In particular for semi-automatic image annotation, additional knowledge inferred from existing ontologies and the associated annotations, could be used for generating coherent keyword assignments, resulting in a good trade-off between annotation work and annotation quality. As the second aspect of our work, we have introduced a method for incorporating users' semantic classification schemes (views) for supporting classical IR by mapping the user's annotation vocabulary onto the system's ontology. In particular, if a set of rules (*mappings*) is available, queries can be adjusted to the users' needs and retrieval objectives.

6

Incorporating a Pseudo Query Reformulation Method for Relevance Feedback in Web Image Retrieval

Relevance feedback (RF) is achieved through users' interaction with the system by evaluating individual result tuples and through the system's query reformulation to better reflect the information need. In this chapter, we present a *Pseudo Query Reformulation* strategy where the iterative computation of relevance values responsible for the reordering of query results is solely based on *relative* distances between images. The particular aspect of our approach is the fact, that the involved functions, like result judgments, relevance computation and reordering of the results, are implemented as database routines (user-defined functions), making our approach highly suitable for web retrieval application. The experimental evaluation demonstrates the effectiveness of the presented relevance feedback approach.

6.1 Introduction

The proceeding application of multimedia information systems and the rapid expansion of image data on the web has brought the need for developing efficient querying and browsing methods for this high-dimensional data. A powerful and widely used technique for improving content-based image retrieval and for narrowing the semantic gap is the relevance feedback method [RHM98, SB90, OBM03, PMO99], which allows query reformulation (QR) by considering the user's subjectivity and perception. In systems supporting this technique the relevance feedback cycle is initialized by users' selection of a set of images that appears to be relevant to an initial query. The subjective user evaluation serves as input for the feedback algorithm which uses the features derived from the selected tuples to revise the search parameters. In general, feedback is used to model the concept the user bears in mind.

As schematized in Fig 6.1, the relevance feedback algorithm is often formulated in terms of the modification of the query vector, adaptation of the similarity metrics [ISF98], or the modification of internal object representation (e.g. in [HZ01]). This cycle of relevance feedback is iterated until the user is satisfied with the retrieved data.

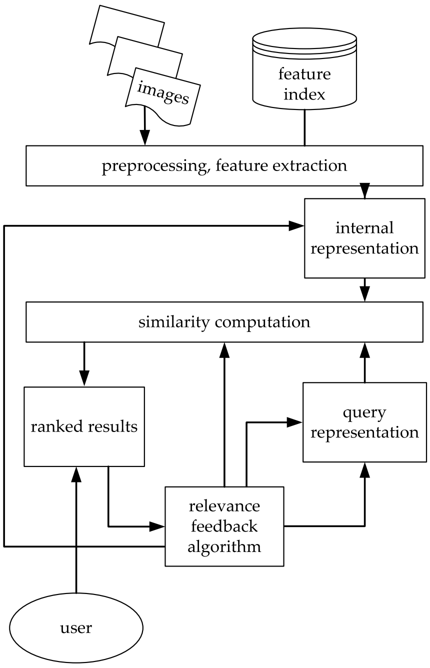


Figure 6.1: Relevance Feedback Cycle in CBIR

This chapter is organized as follows: The remainder of this section reviews different methods for query reformulation and gives the motivation for our 'pseudo' RF approach and presents its specific characteristics. In Section 6.2, we introduce the system's components and the used technologies, like user-defined functions and QBIC's [FSN$^+$95] query-by-content functionalities used for feature extraction and similarity computation between the considered images in the feedback procedure. Section 6.3 demonstrates the implementation details, including functions for the computation of the relevance judgments and methods for updating the scoring of the relevant/irrelevant result tuples. The evaluation on a real world image collection in Section 6.4 demonstrates the behavior of our system and presents the results of the implemented pseudo query refinement. Finally, Section 6.6 gives conclusions and directions for future work.

Relevance Feedback and Pseudo Relevance Feedback

Generally speaking, RF is an iterative process where the initial query is updated at each stage based on the user's feedback. In image retrieval applications the steps include:

1. The user expresses his/her information need by submitting a query q using one of the traditional CBIR paradigms, like *query-by-color*, *query-by-sketch* or *query-by-example.*

2. The system calculates k most similar images $I_1, \ldots, I_k$ to the query image I^q based on their low-level features. This can be performed by applying the k-Nearest Neighbors algorithm [SDI06] which returns a set of images which are similar to the target image I^q and which satisfy the criterion $sim(I_q, I_x) < \varepsilon$, whereas $sim(\cdot)$ $(0 < sim(\cdot) \leq 1)$ computes the similarity between two low-level feature vectors.

3. The user sequentially provides judgments on a limited number of the ranked images from the result set by declaring their relevance or irrelevance to her/his request. These judgments can be related to the individual images as a whole or only to individual features/attributes.

4. The system reformulates the query according to the user's judgments using a particular feedback approach.

5. This cycle of relevance feedback is iterated until the result set reflects the user's information need.

Approaches of reformulating the query can be coarsely divided into *query re-weighting* [WZ02, PMO99], *query representation modification* [RHM97] (see Figure 6.2) and *pseudo relevance feedback* [YHJ03]. All these approaches are based upon the *vector space model* [KSR99] (VSM) from the information retrieval theory [BYRN99, Roc71], according to which images are represented as feature attribute (or weights) vectors in a multidimensional space. The idea of query re-weighting is to learn feature component weights from relevant images (or/and irrelevant images) and to use them for the computation of new parameters for the subsequent query. In contrast, the query modification approach allows users to modify the query point or to refine its representation. An established method for refining the query is given by *query point movement* [RHM97] assuming that there exists an ideal query point which is estimated by the users' feedback judgements. For that purpose, the query point is adopted to move towards the region in the feature space that contains the relevant images (identified as red circles

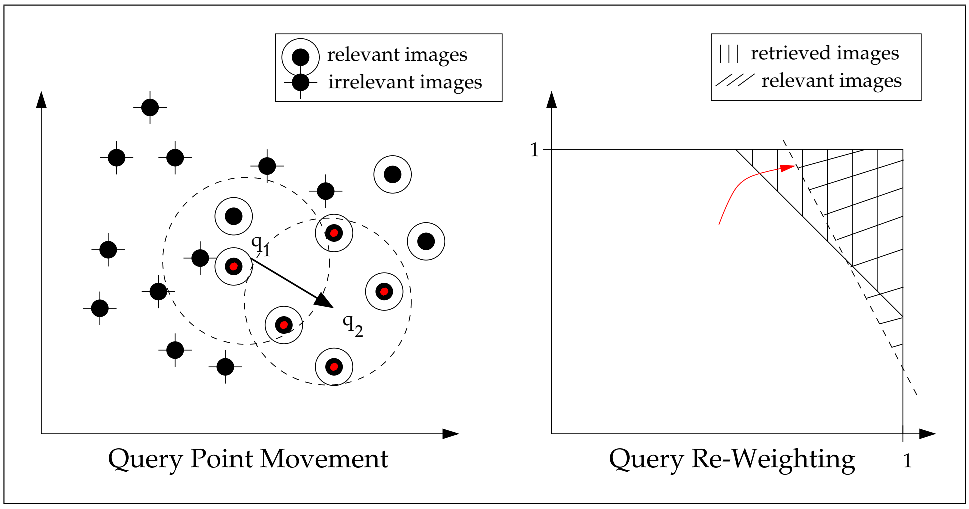


Figure 6.2: Query Point Movement and Query Re-weighting

in Figure 6.2, left) specified by the user and thus approximating best the information need.

A *pseudo feedback approach* (e.g. [YHJ03]) demands minimal participation from the user, because it is based on the assumption, that the k-ranked images in the query result already include relevant images, only their order has to be adjusted according to the user's selection. Hence, the feedback steps are done to judge the relevance of the data, resulting in a reordering of the result set. The choice of k should comprise only a small number of images (e.g. $k = 30$), to avoid displaying a large result set at a time and thus minimizing the users' interaction.

From the users' point of view, the judgment of the result tuples is performed by *explicit* or *implicit* feedback. Giving explicit feedback means that the system captures the documents which were marked as relevant or irrelevant. In contrast, the automatic derivation of what users may be interested in from their behavior is referred to implicit feedback. For example, this information could be inferred from the analysis of documents which have been selected for reading, or how long they have been viewed. Our approach is based on the explicit judgment method with the objective to modify the order of the result set. Since the finer scaling of the relevance values does not significantly influence the results of the feedback algorithm, as shown in empirical studies [JSS00], the users' feedback (UF) is expressed by values [-1, 0, 1] with the following meanings:

$$\text{Users' Feedback Values (UF)} = \begin{cases} -1 & \text{not relevant} \\ \;\;0 & \text{neutral or not evaluated} \\ +1 & \text{relevant} \end{cases} \tag{6.1}$$

The new aspect in our work is the fact that the feedback algorithm is solely based on considering the *relative* distance between images/image features instead of the corresponding feature values commonly used to reformulate the query. This assumption is motivated by the frequent lack of comparison criteria between images lying in different feature spaces and the time-consuming feature extraction which is not recommended for real-time applications. In addition, by only considering the relative distances between images, other distance measures, e.g. for the expression of semantic relationship between two image objects, could be incorporated into our framework without affecting its relevance feedback functionality. Above all, the implementation of the feedback and scoring functionalities as user-defined functions makes the approach primarily suitable for the usage in web-based image retrieval applications.

6.2 System Components

In this section we present the components of our system and the technologies used to perform feature extraction, to capture the information from the feedback and accordingly reorder the result list. The idea to embed relevance feedback procedures into an object-relational database have been proposed in [OBCM02], where the approach has been proven be an effective refinement strategy.

6.2.1 DB2 Image Extender and QBIC

The DB2 database management system provides functionalities for the development of *user-defined types* (UDTs) and *user-defined functions* (UDFs) required for the realization of the feedback functions. For the management and storage of image data, we use the DB2 Image Extender [IBM03] providing similarity search functionalities based on the QBIC [FSN+95] technology for images stored in the `DB2IMAGE` type. The extender is a part of the DB2 AIV Extenders Suite and allows to query image data or search for images based on their content as easily as for traditional textual data [Sto02, IBM03]. Furthermore, new data types and functions for image data using UDTs and UDFs can be created. Another DB2 functionalities can be used in the Image Extender, for example triggers to provide integrity checking across database tables ensuring the referential integrity of image data. An example for inserting an image as the `DB2IMAGE` data type into the table `'Person'` is demonstrated in Figure 6.3.

QBIC complements traditional queries that use image file names or keyword descriptions by *query-by-image-content* functionality. The *QBIC catalog* is a set of administrative support tables that holds data about the visual features of images. An cataloged image is analyzed by the Image Extender by determining its feature values, which are subsequently stored in the QBIC catalog. The *QbScore* describes the distance

```
INSERT INTO Person VALUES(
  ’128557’,           % primary key
  ’Watson’,           % name
    DB2IMAGE(         % DB2IMAGE
      CURRENT SERVER, % server name
      ’watson.jpg’,   % file name
      ’ASIS’,         % do not convert file format
      1,              % save image data as BLOB
      ’chief’)        % comment
);
```

Figure 6.3: Inserting a DB2IMAGE into a Table

between a cataloged image (target image) and a certain feature f_i or (weighted) feature scores $\mathcal{S}(f_{1,x_i}), \ldots, \mathcal{S}(f_{p,x_i})$ of an arbitrary image x_i to compare with. For example, in the weighted case, the score can be computed by

$$QbScore = \frac{\alpha_1}{p} \cdot \mathcal{S}(f_1) + .. + \frac{\alpha_p}{p} \cdot \mathcal{S}(f_p), \tag{6.2}$$

where p is the number of existing features and the values $\alpha_i, \ldots, \alpha_p$ denote the weighting factors for each feature. To return the score of an image, one of these functions has to be called: QbScoreFromString, QbScoreFromTbString, QbScoreFromName, or QbScoreFromTbName. These functions differ from each other by their parameters, the first takes the name of a predefined query as parameter, whereas the second takes the query string directly. As an example, the syntax for the computation of the *weighted score* between the images img1.gif and img2.gif with respect to the available attributes average color, histogram, draw and texture, is introduced in Figure 6.4.

```
SELECT id, name,
  mmdbsys.QbScoreFromStr(img1.gif,
     ’average   file=<server,/pics/img2.gif> weight=2 AND
      histogram file=<server,/pics/img2.gif> weight=0.5 AND
      draw      file=<server,/pics/img2.gif> AND
      texture   file=<server,/pics/img2.gif>’) as QbScore
FROM imagetable;
```

Figure 6.4: Extracting the *QbScore* between img1.gif and img2.gif

In this expression, the weight is a positive real number denoting the significance degree of a particular feature. If no weight has been specified, the default value of 1 is assigned, whereas specifying a weight of zero excludes the respective feature from the computation. In order to determine the distance between images we used QBIC's query-by-image-content functionalities. This distance, the so-called *score* [FSN$^+$95], takes a value between [0,∞] indicating how closely features of an image match those specified in the query. The lower the score, the closer to each other the considered images lie in the feature space.

6.2.2 Similarity Model

Before we consider the realization of query refinement, we first present the feedback and similarity model on which the relevance computation is based upon.

The relevance feedback can be formulated as an optimization problem, with the aim of finding iteratively an optimal query vector $\hat{q}(\vec{p}_{opt})$ with query parameters $\vec{p}_{opt}$ in reference to a initial query $\vec{q}(\vec{p})$ which will provide a result set of relevant images all of them satisfying the user's information need. Hence, the optimization problem can be described by:

$$|\vec{q}(\vec{p}) - \hat{q}(\vec{p}_{opt})| < \epsilon. \tag{6.3}$$

To compute the relevance of a given piece of information with regard to a query, the following similarity model $\mathcal{M}$ is defined:

$$\mathcal{M} = (\textit{attributes, predicates, similarity function}) \tag{6.4}$$

The meaning of the individual terms is demonstrated in the form of a relational query which is presented in the following Example 5.1.

Example 5.1 *Query.*

```
SELECT T.name, weighted_sum(a, 0.4, b, 0.6) AS overall_similarity
FROM Student T
WHERE T.registered AND similar_marks(T.mark, 2, "Databases", 0.5, a)
AND live_close_to(T.city, "Düsseldorf", 0.5, b)
ORDER BY overall_similarity ASC;
```

The presented query finds all the names of the students that are registered, have good marks in the subject *Databases* und live close to the city *Düsseldorf*. The query has two similarity predicates: *similar_marks* and *live_close_to*, which return two similarity values `a` and `b`. These two values are combined into a single `overall_similarity` by a similarity function (*weighted_sum*). Formally, for a given list of similarity values $s_1, \ldots, s_n$ ($s_i \in [0,1]$) and a corresponding weight $w_1, \ldots, w_n$ ($w_i \in [0,1]$ and $\sum_i w_i = 1$) for each s_i value, the similarity function has the form:

$$\textit{similarity\_function}(s_1, w_1, \ldots, s_n, w_n) \rightarrow [0,1]\,. \tag{6.5}$$

The similarity predicates *similar_marks* and *live_close_to* are functions with freely definable number of input values. Here, the first value is the attribute to compare, followed by function specific values, and the last two values include the threshold α and the similarity score as return value. The function returns true if the similarity $> \alpha$, else it returns false. Depending on the situation, the number of input values may be adapted to the required computation, for example some parameters may be added to distinguish between different distance models or to configure the functions.

6.2.3 Feedback Algorithm

Since it is barely impossible for users of an image retrieval system to formulate the query as a sequence of SQL statements containing scoring functions (e.g. similarity functions for particular features) and required parameters (like feature weights), the internal computations have to be embed in a query refinement strategy.

The basis for the implemented refinement is Rocchio's formula [Roc71], which formulates the query point movement iteratively approximating the ideal query point. This is done by moving the query towards relevant points (documents which have been marked as relevant by the user) and away from non-relevant points. The Rocchio's formula is given below for a set of relevant documents D^+ and non-relevant documents D^-:

$$Q' = \alpha\, Q + \beta\, (\frac{1}{|D^+|} \sum_{i \in D^+} d_i) - \gamma\, (\frac{1}{|D^-|} \sum_{i \in D^-} d_i) \tag{6.6}$$

where α, β, and γ are suitable constants which are determined by heuristics.

In our algorithm, the query point movement operates on image data represented by all features available in QBIC. Let $\mathcal{U}$ be the *universe* of images and let $C \subset \mathcal{U}$ be a fixed, finite collection of images. For a given query q, the user has in mind some relevant set of images $\mathcal{I}_q^+ \subset C$. This set is unknown and the system's objective is to discover in optimal case all of these images. The interactive retrieval process starts with the user proposing a particular query image, $I^q \subset \mathcal{U}$. Then the system provides an initial set $\mathcal{I}_q \subset C$ of k images that are similar to I^q according to a suitable distance measure. This set of images is judged by the user who provides feedback values presented in the formula 6.1 by marking images as relevant or not relevant. Now, this feedback information is used by the system to recompute a new set of images and the process is repeated until the user is satisfied with the results.

Since most of the web retrieval systems, and also the used QBIC system, do not reveal the internal representation of the data, the modification of the query has to be done by considering the feature distances (scores) of the images which have been evaluated by the user. This set of images serves as multiple examples which are used to determine the overall score (distance) to the optimal query, and by means of this score, certain images are added or excluded in the next feedback iteration.

In the initial user's query, all available features in QBIC are considered for the score computation. The scores $\mathcal{S}(f_{p,x_i})$ are stored in the feedback table (see Figure 6.6) for each feature f_p and image x_i in the data collection. These values are used to create a ranked result list which is subsequently presented to the user. This list is ordered by the overall score. In the first user's feedback, the result list is examined and certain features are given a feedback value introduced in Equation 6.1. The new (overall) score $\mathcal{S}$ is calculated as follows:

$$\mathcal{S}(f_{p,x_i})^{\text{new}} = \alpha \cdot \mathcal{S}(f_{p,x_i})^{\text{old}} + \frac{\beta}{n} \cdot \sum_{i=1}^{n} s(f_{p,x_i^+}) + \frac{\gamma}{m} \cdot \sum_{j=1}^{m} (\max(f_{p,x_j^-}) - s(f_{p,x_j^-})), \quad (6.7)$$

where $\mathcal{S}(f_{p,x_i})$ denotes the score of the image x_i with respect to the query and the feature f_p. The scores $s(f_{p,x_i^+})$ and $s(f_{p,x_i^-})$ are computed for images in the result set which have been given a positive (images $x_i^+ = x_1, \ldots x_n$) or negative feedback (images $x_i^- = x_1, \ldots x_m$). n and m present the number of feedbacks which were positive or negative, and $\max(f^-_{p,x_j})$ is the maximum value of f_p, occurring in the negative examples. The parameters α, β, and γ are used to describe the influence of the previous iteration or the influence of the negative and positive feedback. According to [Roc71], the conditions $\beta > \gamma$ and $\alpha + \beta + \gamma = 1$ have to be fulfilled. For example, each time an image is judged as relevant for a particular query in respect to feature f_p (e.g. $f_1 = \mathit{color}$), its score $s(f_{p,x_i})$ is computed. As summary, Equation 6.7 presents the computation of the overall score $\mathcal{S}(f_{p,x_i})$ for a given feature f_p. It is determined by averaging the gathered individual scores $s(f_{p,x_i^+})$ of images which have been marked as relevant and the weighted attenuation of the score if the feature also occurred in images which have been marked as irrelevant.

6.3 Implementation Details

The implementation of the relevance feedback focusses on three main procedures, each containing several functionalities (Figure 6.5).

A. Initialization of the Feedback. The first procedure `initFeedback()`, parameterized with an initial query image, is called to initialize the retrieval/feedback loop and to reset all auxiliary tables.

B. Execution of the Feedback. The `feedback()` procedure is invoked for every image, which has been evaluated by the user, providing scores to be buffered in the auxiliary tables.

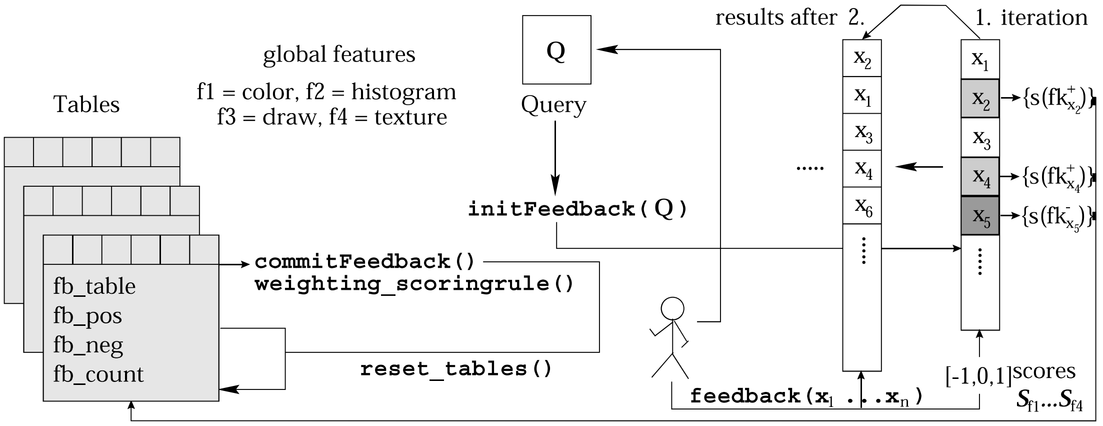


Figure 6.5: Realization of the Feedback Methods

C. **Commit the Feedback.** After the specification of the parameters α, β, and γ, the feedbacks are calculated using the procedure `commitFeedback(`α, β, γ`)`.

The computed scores for each image and other auxiliary values are stored in temporary tables schematized in Figure 6.6, each consisting of the key attribute `tid` and four attributes for the individual feature scores. Table `fb_table`, standing for 'feedback table' stores the currently valid overall score for each image and feature (determined by applying the formula 6.7), whereas tables `fb_pos` and `fb_neg` collect the individual scores from the positive and negative user feedbacks. For example, if image x_2 gets a negative feedback, the scores $s(f_{p,x_2})$ are computed for the four features $f_1, \dots, f_4$ and the tuple $(x_2, s(f_{1,x_2}), s(f_{2,x_2}), s(f_{3,x_2}), s(f_{4,x_2}))$ is inserted into the table `fb_neg`. Another auxiliary table, like `fb_temp`, is required for buffering the results and table `fb_count` provides a counter for the executed positive and negative feedbacks.

feedback: `(fb_table)`

`tid`	$f1$	$f2$	$f3$	$f4$
x_1	$\mathcal{S}(f_{1,x_1})$	$\mathcal{S}(f_{2,x_1})$	$\mathcal{S}(f_{3,x_1})$	$\mathcal{S}(f_{4,x_1})$
x_2	$\mathcal{S}(f_{1,x_2})$	$\mathcal{S}(f_{2,x_2})$	$\mathcal{S}(f_{3,x_2})$	$\mathcal{S}(f_{4,x_2})$
x_3	$\mathcal{S}(f_{1,x_3})$	$\mathcal{S}(f_{2,x_3})$	$\mathcal{S}(f_{3,x_3})$	$\mathcal{S}(f_{4,x_3})$
...	...	...	...	...

positive feedback: `(fb_pos)`

`tid`	$f1$	$f2$	$f3$	$f4$
x_1	0.1	0.2	0.0	0.0
x_2	0.0	0.5	0.0	0.0
x_3	0.3	0.2	0.0	0.0
...	...	...	...	...

negative feedback: `(fb_neg)`

`tid`	$f1$	$f2$	$f3$	$f4$
x_1	0.1	0.0	0.0	0.2
x_2	0.5	0.0	0.0	0.2
x_3	0.2	0.0	0.0	0.7
...	...	...	...	...

Figure 6.6: Tables used for the Query Refinement

A. Preparing the Database – initFeedback():

In this first step, `initFeedback()` is invoked with the query image's file name. Afterwards, the zero filled temporary tables are created by taking the template table `feedback` and are subsequently filled with the initial distances computed by QBIC.

```
 CREATE PROCEDURE initFeedback
   (IN imagename VARCHAR (15))
   NO EXTERNAL ACTION
   LANGUAGE SQL
 BEGIN ATOMIC
%%% Creating zero-filled temporary tables
%%% Initializing the QBE-Image
%%% Inserting the initial distances

 INSERT into session.fb_table (
   SELECT tid,
      mmdbsys.QbScoreFromStr(image,
     'QbColorFeatureClass file = X) as a1,
      mmdbsys.QbScoreFromStr(image,
     'QbColorHistogramFeatureClass file = X) as a2,
      mmdbsys.QbScoreFromStr(image,
     'QbDrawFeatureClass file = X) as a3,
      mmdbsys.QbScoreFromStr(image,
     'QbTextureFeatureClass file = X) as a4
   FROM imagetable);
 END;
```

In the `INSERT` statement of the procedure, the *initial* feature distances between the query image and all images in the data collection (in table `imagetable`) are computed and inserted into the feedback table. In practice, the variable X is replaced by the path of the query image, for example by the expression `<server,/pics/'||imagename||'>'`.

B. Gathering the Feedback – feedback():

To realize the procedure `feedback()` a few help functions have been implemented at first. The user-defined functions `uf_Color`, `uf_HColor`, `uf_Draw`, and `uf_Texture` serve to capsulate the complex score computation and provide a 'user view' for only retrieving the relative *FeatureScore* with respect to the query image and a selected feature. An example of the function `uf_Color` which returns a ranked list of distances between all stored images and a query image `'imagename'` is given above:

```
   CREATE function uf_Color
     (imagename VARCHAR(15))
     RETURNS TABLE (tid INTEGER, f1 decimal(7,3))
     LANGUAGE SQL
   RETURN
    SELECT tid,
     mmdbsys.QbScoreFromStr(image,
       'QbColorFeatureClass file =
       <server,/pics/'||imagename||'>')
    FROM imagetable;
```

The chosen image id which was given a user's feedback, is passed with its relevance value to the feedback function `feedback()`. The relevance values assume the values 1, 0, or -1. Depending on the assigned value for each feature, the functions fb_pos() or fb_neg() are called. The temporary table fb_temp is needed to save the results and to store them in the scoring table. After each successful update of the scores, the counter in table fb_count is increased. In this table, the number of given positive and negative ratings is stored for each feature, thereinafter required for the relevance computation of a particular feature (see Subsection **C.**).

C. Evaluating the Feedback – commitFeedback():

The function scoringRule(), which is responsible for the reordering of the results is presented below. Alternatively, another feedback mechanism can be easily embedded at this place to implement different approaches. The overall score of a respective image is determined using the Formula 6.7 by considering the positive and negative feedbacks, their frequency, and the parameters α, β, γ.

```
CREATE FUNCTION scoringRule(
  a DECIMAL(7,3), b DECIMAL(7,3), c DECIMAL(7,3),
  alpha DECIMAL(7,3), beta DECIMAL(7,3), gamma DECIMAL(7,3),
  max DECIMAL(7,3))
    RETURNS DECIMAL(7,3)
    LANGUAGE SQL
    CONTAINS SQL
    NO EXTERNAL ACTION
    NOT DETERMINISTIC
BEGIN ATOMIC
  IF (a IS NULL AND b IS NULL)
    THEN RETURN c;
  ELSEIF a IS NULL
    THEN RETURN ((alpha+(beta/2)) * c) + ((gamma+(beta/2)) * (max-b));
  ELSEIF b IS NULL
    THEN RETURN ((alpha+(gamma/2)) * c) + ((beta+(gamma/2)) * a);
  ELSE RETURN (alpha * c) + (beta * a) + (gamma * (max-b));
  END IF;
END;
```

With commitFeedback() the gathered feedback values are updated in the temporary tables (fb_pos and fb_neg) after each feedback step, and subsequently reset for the next iteration. The following code fragment demonstrates the subsequent update of the $f1 \ldots f4$ values in table fb_pos by averaging the sum of the gathered scores by the number of positive user's feedbacks for each of the individual features:

```
UPDATE session.fb_pos SET (f1, f2, f3, f4) =
   ( f1 / (SELECT pos FROM session.fb_count
           WHERE f = 1),
      ...
   );
```

In the last step, the function scoringrule() is used to determine and buffer the new scores for all available images according to the existing feedback values. These

```
INSERT INTO session.fb_temp (
  SELECT a.tid as tid,
    scoringrule(a.f1, b.f1, c.f1, alpha, beta, gamma),
    scoringrule(a.f2, b.f2, ...),
    ...
    FROM (fb_pos NATURAL JOIN fb_neg NATURAL JOIN fb_table))
```

scores are inserted into the temporary table `fb_temp` (see `INSERT` statement above). After the numerous steps of this one iteration, the ranked results are reordered in a descending order according to the images' overall scores.

6.4 Experiments and Evaluation

Since the focus of this paper lies on the integration of a pseudo relevance feedback functionality into an object-relational database, and not on the optimization of existing relevance feedback approaches, our evaluation data set only comprised 1,052 images. The enhancement of the retrieval quality was measured by *precision* and *recall*, which were plotted at each feedback iteration, indicating the amount of relevant documents in the result list (precision) and the percentage of relevant documents already found (recall). At each iteration the user gave his relevance judgment to two chosen images. For each of the selected images, the invocation of the sequence of functions presented in Figure 6.7 was necessary to realize the query reformulation. The implemented graphical web interface which provides the possibility to define a query image and to make judgments about the relevance of each feature/image in the result set is presented in Figure 6.8.

```
call initfeedback('imagename');
   For (#relfeed)
       use SESSION.fb_table to call the values
   For (#judgments)
       call feedback('image_x', 1, -1, 0, 1);
   End judgments
 call commitfeedback(0.5, 0.4, 0.1);
End RF cycles;
```

Figure 6.7: Function Calls to Commit Users' Feedback

The experiment consisted of two queries performed by two different users. The first query session (Figure 6.8, left) started with a query-by-sketch using a two-colored image template as q_1 simulating a sunset. This special scenario was intentionally chosen due to its outstanding reproduction of the feature *color* and to simulate the position, that searchers often have no idea what they are looking for. In the second query session (see Figure 6.8, right), a grey scale image of a building was taken as the start point

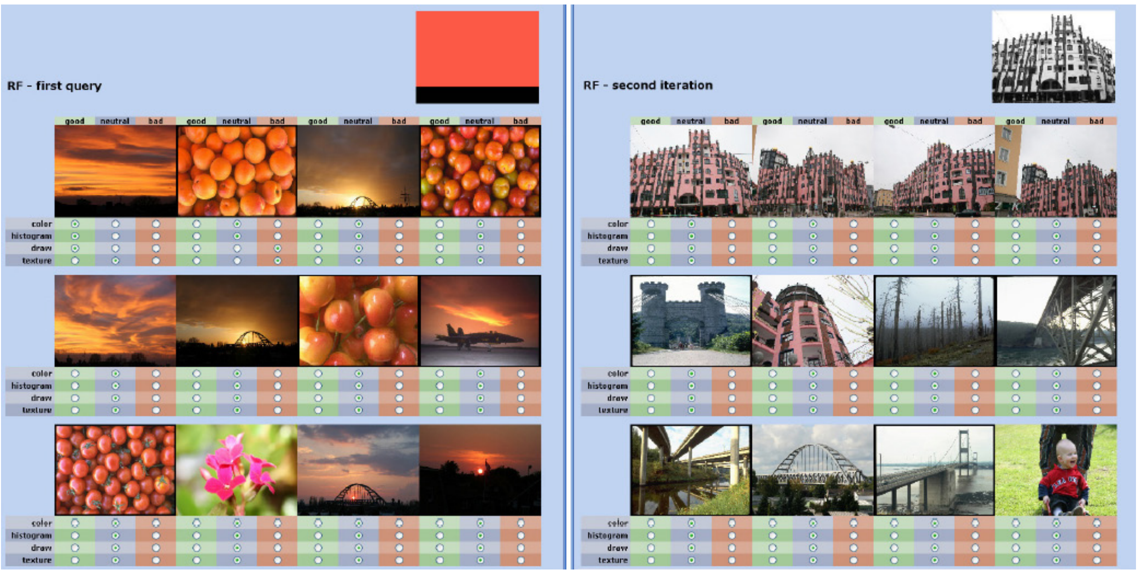


Figure 6.8: Graphical Web Interface for Retrieval and Relevance Feedback. The Displayed Images Present the Initial Results of the Queries q_1 and q_2

q_2, with its prominent texture properties. The results obtained from the experiments were analyzed from several aspects:

a.) effectiveness of the similarity functions of QBIC,

b.) relevance feedback effort, e.g. number of images viewed, duration of the judgments,

c.) subjective evaluation of the usefulness of the refined answer set after first and second RF iteration.

Figure 6.9 shows the precision versus recall curves for the two query sessions, initiated with queries q_1 and q_2. Considering the manageable amount of four available low-level features and the limited image application domain, the CBIR functionalities of QBIC provided adequate results for the subsequent relevance feedback evaluation. Since the number of displayed result tuples was limited to 9 and the judgment of the images was executed rather efficiently, the time factor could be neglected in this evaluation. The curve progression in both cases shows a high performance of the two queries already after the initial query, but could be increased after the first and second feedback iteration.

In addition, the users' subjective feeling about the usefulness of the query reformulation showed that the reordering of the top-ranking images after the first and second feedback iterations fulfilled its requirements. The new determined images appeared to be a natural expansion of the initial query submitted by the user.

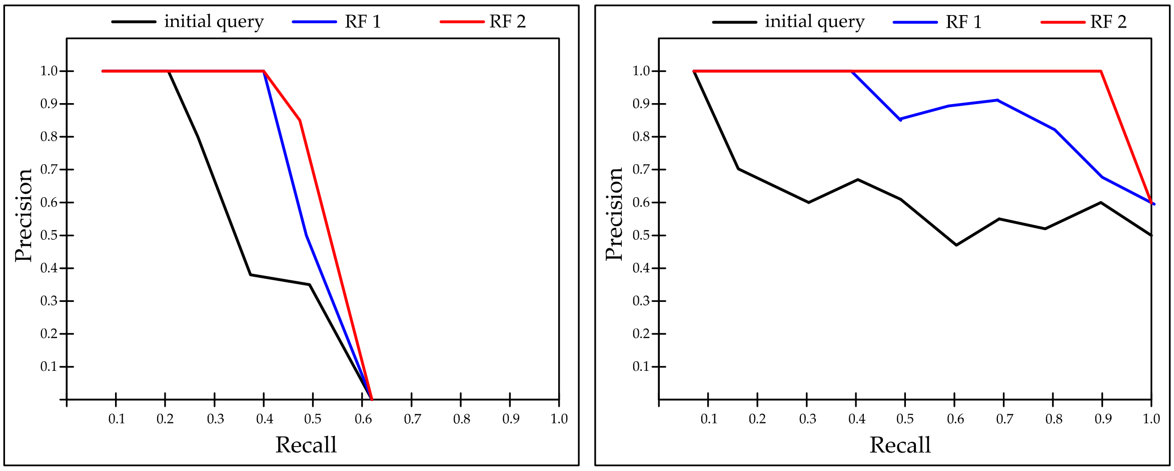


Figure 6.9: Precision/Recall Curves for the Queries q_1 and q_2 After First and Second RF Iteration

6.5 Related Work

Our approach is related to numerous work from several areas of multimedia IR, like CBIR, query refinement, relevance feedback and query formulation. The embedding of relevance feedback procedures as functions into an object-relational database was inspired by the work of [OBCM02]. In our contribution, we do not focus on the optimization of query refinement strategies or the improvement of similarity functions for image data, but rather, we work out, how to incorporate a pseudo relevance feedback method into web retrieval applications which will encapsulate the internal feature extraction functionalities from the user. Hence, our feedback algorithm is solely based on the *relative* distance between images/image features instead of the corresponding feature values commonly used to reformulate the query. Traditionally, similarity computation and relevance feedback have been studied for textual data and have been recently generalized to other application fields, like images [HROM98, Pen03], temporal data [KP99], or web retrieval [YCWM03]. Some representative systems using the relevance feedback for CBIR are MARS [RHM97] and Photobook [PPS99]. MARS implements a single-point movement technique, which means that the refined query q at each iteration consists of only one query point. By contrast, multi-point movement techniques, such as query expansion [COBMP04] or Qcluster [KC03], use multiple query points to estimate the ideal space that is most likely to contain relevant results. Experimental evaluation in [RHM97] shows that query expansion outperforms query point movement in retrieval effectiveness. Another advantage of query expansion is that query expansion can be coupled with existing information systems without requiring any modification of the internal query representation.

In last years, several extensions of the classical RF approaches have attracted research communities. For example, MediaNet [HRTL04] is an approach which integrates additional knowledge sources into the relevance feedback process and uses genetic or evolutionary algorithms directly for the search process. The additional knowledge sources are used to shape the learning space when insufficient training samples are available. In web image retrieval applications the RF have been avoided so far because of scalability, efficiency and effectiveness reasons. In [CJZJ06] a combination of visual feature-based RF and textual feature-based RF mechanism was proposed, which collects the implicit click-through data without extra burden on the user. Since web images could be characterized by textual and visual features, the use of textual features can be beneficial to image retrieval by incorporating high-level concepts. In our strategy, the query response time of queries could be negligible, and by the restriction of the initial result set, and thus the number of user's explicit interactions, we could achieve that relevant images could be found without any effort.

6.6 Conclusions

In this chapter we have presented a framework for incorporating a pseudo relevance feedback procedure for image retrieval using IBM DB2 and the QBIC system. The similarity computation, result judgements and query refinement have been integrated into the SQL language by using procedures and user-defined functions. A final evaluation of the result quality has been done to validate the approach taken. In summary, the results provide a solid basis for further research activities. Particularly in cases when there is no adequate query image as initial point, we can achieve a significant increase of the retrieval quality by the implemented pseudo relevance feedback procedure. As mentioned in the motivation, another promising research direction could be to combine the low-level similarity with high-level relations between semantic concepts. For example, the extraction of semantic information could be automated (e.g. in web retrieval applications) by considering the bounding textual information around the image data. Furthermore, our approach could be combined with additional knowledge in form of domain-specific ontologies and thus provide support for manual semantic classification of the data. From this classification, knowledge about the user's perception subjectivity could be inferred and utilized for the relevance feedback.

7

Conclusion and Future Work

This chapter presents the conclusion of our work. Section 7.1 summarizes the contributions of this thesis and describes the solution of the given problems. Finally, some future research directions are presented in Section 7.2.

7.1 Summary

The late advances in computer and communication technologies caused a huge increase of digital multimedia information available in personal and business applications. Several new requirements for satisfying the users' needs during the retrieval and annotation of multimedia data which have appeared due to this development, have been considered in this thesis. First of all, we have presented a framework for supporting semi-automatic annotation of multimedia data which is based on the extraction of elementary low-level features, user's relevance feedback, and the usage of ontology knowledge. This approach facilitates image annotation by computing the most likely relevant content descriptors as a result of extracted low-level features and the comparison of annotations of similar images. Besides the definitions used throughout this thesis and the detailed description of the image's representation levels, we have considered the levels at which relevance feedback is applied within our framework. In addition, we have supplementary focused on the projection of visual features into a finite set of semantic concepts which stills forms a real challenge in retrieval applications.

Another aspect of our work results from the encountered problems during the annotation process, like the existence of multiple levels of abstraction, incompleteness of annotation data, or differing users' subjectivity. We have firstly introduced fundamental definitions needed for the introduction of the multi-level annotations. Within our Annotation Analysis Framework, a graph-based representation technique is used to transform the annotations into a form which is understandable for the machine by facilitating inference making. The presented method incorporates the semantic meaning od annotation terms, their relations, and the frequency they are assigned, and thus supports semantic retrieval at different levels of abstraction. In addition, we have demonstrated how to incorporate our method into the probabilistic image annotation approach.

In order to avoid context mismatches between users, for example when users' preferences, linguistic differences, or the usage of different abstraction levels for the annotation influences retrieval behavior of an IR system, methods for understanding and interpreting the subjective sights are needed. Based on our annotation/retrieval framework, we have presented the GLENARVAN component, which is responsible for context computation, ontology comparison, and query expansion according to users' profiles. In this contribution we have considered two different aspects: First, multiple sources of information which are modeled as different user profiles and are brought together in order to extract contextual information and to attenuate users' subjectivity. The second issue is how to prevent the retrieval process to fail in the case of different views on the data collection. For this purpose, the subjective users' annotations are used to discover mappings between the system's ontology and the user's vocabulary and thus to infer additional query parameters for a user-adapted query reformulation.

Finally, we have presented a Pseudo Relevance Feedback method, which improves the content-based image retrieval by query reformulation considering the user's subjectivity and perception. The feedback cycle is characterized by users' interaction with the system in which individual result tuples are evaluated as relevant or not relevant for a given query. The particular aspect of our approach is the fact, that the involved functions, like result judgments, relevance computation and reordering of the results, have been implemented as user-defined functions, making the method highly suitable for web retrieval applications. The subsequent experimental evaluation on an image collection demonstrates the effectiveness of the presented relevance feedback approach.

7.2 Future Work

In the context of this thesis we have focused on a small set of possible functionalities to improve the semantic multimedia retrieval. However, referring to the concepts we have presented in this work, there are several aspects that would require further investigations: The extraction of primitive low-level features (pixel-based extraction) has some limitations that need further considerations. A question could be here, to investigate the impact of feature selection on the performance of the semi-automatic annotation, since all hybrid approaches depend on the performance of CBIR algorithms.

Another promising aim is the improvement of the annotation quality, which presents an important requirement for annotation-based retrieval systems or systems performing (semi-)automatic assignment of annotations. The GLENARVAN component could be expanded by a data generator component, transforming the analyzed annotation behavior of a user (profiles) and the used vocabulary into training data. The captured information retained over multiple system interactions together with the mappings between different annotation profiles could be profitable for systems which are based on machine learning. Particularly in systems, which are based on the automatic recommendation of suitable annotations for a given image, the training data may be used for providing coherent keyword assignments, and in the end, this would result in a good trade-off between annotation work and annotation quality.

References

[AAB$^+$03] J. Allan, J. Aslam, N. Belkin, C. Buckley, J. Callan, B. Croft, and et al. Dumais. Challenges in Information Retrieval and Language Modeling: Report of a Workshop Held at the Center for Intelligent Information Retrieval, University of Massachusetts Amherst. *SIGIR Forum*, 37(1):31–47, 2003.

[AAR04] S. Agarwal, A. Awan, and D. Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[AB94] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 94–105, New York, NY, USA, 1998. ACM Press.

[ASSB05] P. Appan, B. Shevade, H. Sundaram, and D. Birchfield. Interfaces for Networked Media Exploration and Collaborative Annotation. In *IUI '05: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 106–113, New York, NY, USA, 2005. ACM Press.

[AT97] F. Asnicar and C. Tasso. ifWeb: a Prototype of User Model-based Intelligent Agent for Documentation Filtering and Navigation in the Word Wide Web. In *Proceedings of the 1st International Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 3–12, 1997.

[BBFH02] J. Budzik, S. Bradshaw, X. Fu, and K. J. Hammond. Supporting On-Line Resource Discovery in the Context of Ongoing Tasks with Proactive Software Assistants. *International Journal of Human-Computer Studies*, 56(1):47–74, 2002.

[BBH$^+$87] A. Blumer, J. Blumer, D. Haussler, R. McConnell, and A. Ehrenfeucht. Complete Inverted Files for Efficient Text Retrieval and Analysis. *Journal of the ACM*, 34(3):578–595, 1987.

[BBK01] C. Böhm, S. Berchtold, and D. A. Keim. Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases. In *ACM Computing Surveys*, volume 33, pages 322–373, New York, NY, USA, 2001. ACM Press.

[BDPDPM93] P. E. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. In *Computational Linguistics*, volume 19, pages 263–311, Cambridge, MA, USA, 1993. MIT Press.

[BG00] D. Brickley and R. V. Guha. Resource Description Framework (RDF) Schema Specification. World Wide Web Consortium. `http://www.w3.org/TR/rdf-schema`. 2000.

[BGG$^+$99] D. Boley, M. Gini, R. Gross, E.-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document Categorization and Query Generation on the World Wide WebUsing WebACE. *Artificial Intelligence Review*, 13(5-6):365–391, 1999.

[BGP03] J. Bormans, J. Gelissen, and A. Perkis. MPEG-21: The 21st Century Multimedia Framework. *IEEE Signal Processing Magazine*, 20(2):53–62, 2003.

[BH04] T. R. Bruce and D. Hillmann. *The Continuum of Metadata Quality: Defining, Expressing, Exploiting*, pages 238–256. ALA Editions, Chicago, IL, 2004.

[BKK96] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: An Index Structure for High-Dimensional Data. In *VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases*, pages 28–39, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[BLHL01a] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. In *Scientific American*, volume 284, pages 28–37, 2001.

[BLHL01b] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities. *Scientific American*, 279(5):34–43, 2001.

[BM02] P. Brusilovsky and M. T. Maybury. From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM*, 45(5):30–33, 2002.

[Bru96] P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996.

[BS95] C. Buckley and G. Salton. Optimization of Relevance Feedback Weights. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'95*, pages 351–357, New York, NY, USA, 1995. ACM Press.

[BS01] A. B. Benitez and J. R. Smith. New Frontiers for Intelligent Content-Based Retrieval. *In Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases (IS&T/SPIE)*, 4315:141–152, 2001.

[BSA94] C. Buckley, G. Salton, and J. Allan. The Effect of Adding Relevance Information in a Relevance Feedback Environment. In *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 292–300, New York, NY, USA, 1994. Springer-Verlag.

[BYRN99] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.

[Cam02] D. G. Campbell. The Use of the Dublin Core in Web Annotation Programs. In *DCMI '02: Proceedings of the 2002 International Conference on Dublin Core and Metadata Applications*, pages 105–110. Dublin Core Metadata Initiative, 2002.

[CBGM02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[CC03] P.-J. Cheng and L.-F. Chien. Effective Image Annotation for Search using Multi-level Semantics. In *Proceedings of International Conference of Asian Digital Libraries*, pages 230–242. Springer-Verlag, 2003.

[CCH92] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY Retrieval System. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[CCS03] C. Cusano, G. Ciocca, and R. Scettini. Image Annotation using SVM. In *Proceedings of Internet Imaging V*, volume 5304, pages 330–338. The International Society for Optical Engineering (SPIE), 2003.

[CFPS02] D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic Query Refinement using Lexical Affinities with Maximal Information Gain. In *SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 283–290, New York, NY, USA, 2002. ACM Press.

[CH79] W. B. Croft and D. Harper. Using Probabilistic Models of Document Retrieval without Relevance Information. In *Journal of Documentation*, volume 35, pages 285–295, 1979.

[Cit] Citeseer, Scientific Literature Digital Library. `http://citeseer.ist.psu.edu/`.

[CJZJ06] E. Cheng, F. Jing, L. Zhang, and H. Jin. Scalable Relevance Feedback using Click-Through Data for Web Image Retrieval. In *MULTIMEDIA '06: Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 173–176, New York, NY, USA, 2006. ACM Press.

[CLC06] J. Chakravarthy, V. Lanfranchi, and F. Ciravegna. Community-based Annotation of Multimedia Documents. In *Poster Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*, 2006.

[CM00] K. Chakrabarti and S. Mehrotra. Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces. In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 89–100, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[COBMP04] K. Chakrabarti, M. Ortega-Binderberger, S. Mehrotra, and K. Porkaew. Evaluating Refined Queries in Top-k Retrieval Systems. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):256–270, 2004.

[CSB$^+$03] J. Callan, A. Smeaton, M. Beaulieu, P. Borlund, P. Brusilovsky, M. Chalmers, C. Lynch, J. Riedl, B. Smyth, U. Straccia, and E. Toms. Personalisation and Recommender Systems in Digital Libraries. In *Joint NSF-EU DELOS Working Group Report*, 2003.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: and Other Kernel-based Learning Methods.* Cambridge University Press, New York, NY, USA, 2000.

[CTB+99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A System for Region-Based Image Indexing and Retrieval. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 509–516, London, UK, 1999. Springer-Verlag.

[CV05] G. Carneiro and N. Vasconcelos. Formulating Semantic Image Annotation as a Supervised Learning Problem. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 163–168, Washington, DC, USA, 2005. IEEE Computer Society.

[DBdFF02] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag.

[DMDH02] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to Map between Ontologies on the Semantic Web. In *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, pages 662–673, New York, NY, USA, 2002. ACM Press.

[DR01] K. R. Debure and A. S. Russell. Feature Extraction for Content-based Image Retrieval in DARWIN. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, page 470, New York, NY, USA, 2001. ACM Press.

[Fli] Flickr. `http://www.flickr.com/`.

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. In *Computer*, volume 28, pages 23–32, Los Alamitos, CA, USA, 1995. IEEE Computer Society.

[GBYS92] G. H. Gonnet, R. A. Baeza-Yates, and T. Snider. New Indices for Text: PAT Trees and PAT Arrays. In *Information Retrieval: Data Structures and Algorithms*, pages 66–82, Upper Saddle River, NJ, USA, 1992. Prentice-Hall, Inc.

[GIK05] A. Ghoshal, P. Ircing, and S. Khudanpur. Hidden Markov Models for Automatic Annotation and Content-Based Retrieval of Images and Video. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 544–551, New York, NY, USA, 2005. ACM Press.

[GMY93] J. Griffioen, R. Mehrotra, and R. Yavatkar. An Object-oriented Model for Image Information Representation. In *In CIKM '93: Proceedings of the Second International Conference on Information and Knowledge Management*, pages 393–402, New York, NY, USA, 1993. ACM Press.

[GMY95] J. Griffioen, R. Mehrotra, and R. Yavatkar. A Semantic Data Model for Embedded Image Information. In *In IS&T/SPIE Symposium on High-Speed Networking and Multimedia Computing*, pages 393–402, 1995.

[GR95] Venkat N. Gudivada and Vijay V. Raghavan. Content-Based Image Retrieval Systems. In *Computer*, volume 28, pages 18–22, Los Alamitos, CA, USA, 1995. IEEE Computer Society Press.

[GRV96] V. N. Gudivada, V. V. Raghavan, and K. Vanapipat. A Unified Approach to Data Modeling and Retrieval for a Class of Image Database Applications. In *Multimedia Database Systems: Issues and Research Directions*, pages 37–78, USA, 1996. Springer-Verlag.

[GS00] W. I. Grosky and P. L. Stanchev. An Image Data Model. In *VISUAL '00: Proceedings of the 4th International Conference on Advances in Visual Information Systems*, pages 14–25, London, UK, 2000. Springer-Verlag.

[GWJ91] A. Gupta, T. Weymouth, and R. Jain. Semantic Queries with Pictures: The VIMSYS Model. In *VLDB '91: Proceedings of the 17th International Conference on Very Large Data Bases*, pages 69–79, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[GZCS94] Y. Gong, H. J. Zhang, H. C. Chuan, and M. Sakauchi. An Image Database System With Content Capturing and Fast Image Indexing Abilities. In *IEEE International Conference on Multimedia Computing and Systems*, pages 121–130, May 1994.

[HKM$^+$97] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 762–768, Washington, DC, USA, 1997. IEEE Computer Society.

[HR01] D. Hiemstra and S. Robertson. Relevance Feedback for Best Match Term Weighting Algorithms in Information Retrieval. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, pages 37–42, 2001.

[HROM98] T. Huang, Y. Rui, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 25–36, 1998.

[HRTL04] M. Haas, J. Rijsdam, B. Thomee, and S. M. Lew. Relevance Feedback: Perceptual Learning and Retrieval in Bio-computing, Photos, and Video. In *MIR '04: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 151–156, New York, NY, USA, 2004. ACM Press.

[HSWW03] L. Hollink, G. Schreiber, J. Wielemaker, and B. Wielinga. Semantic Annotation of Image Collections. In *KCAP '03: Workshop on Knowledge Markup and Semantic Annotation*, 2003.

[HWGS$^+$05] C. Halaschek-Wiener, J. Golbeck, A. Schain, M. Grove, B. Parsia, and J. Hendler. Photostuff - An Image Annotation Tool for the Semantic Web. In *4th International Semantic Web Conference*, 2005.

[HZ01] T. S. Huang and X. S. Zhou. Image Retrieval with Relevance Feedback: From Heuristic Weight Adjustment to Optimal Learning Methods. In *ICIP 2001: Proceedings of the International Conference on Image Processing*, pages 2–5. IEEE Computer Society, 2001.

[IBM03] IBM. *DB2 Universal Database Image, Audio, and Video Extenders Administration and Programming Version 8.* IBM Corporation, 2003.

[Ino04] Masashi Inoue. On the Need for Annotation-based Image Retrieval. In *IRiX '04: Workshop on Information Retrieval in Context*, pages 44–46, Sheffield, UK, 2004.

[Iok89] M. Ioka. A Method of Defining the Similarity of Images on the Basis of Color Information. In *Technical Report, IBM Research, Tokyo Research Laboratory*, 1989.

[ISF98] Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Querying Databases Through Multiple Examples. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages

218–227, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[JLM03] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 119–126, New York, NY, USA, 2003. ACM Press.

[Jol86] I. T. Jolliffe. Principal Component Analysis. *Springer-Verlag*, 1986.

[JSS00] B. J. Jansen, A. Spink, and T. Saracevic. Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web. *Information Processing and Management: an International Journal*, 36(2):207–227, 2000.

[KAAS99] K. V. R. Kanth, D. Agrawal, A. E. Abbadi, and A. Singh. Dimensionality Reduction for Similarity Searching in Dynamic Databases. *Computer Vision and Image Understanding*, 75(1-2):59–72, 1999.

[KBC04] M. Kokare, P. K. Biswas, and B. N. Chatterji. Rotated Complex Wavelet based Texture Features for Content Based Image Retrieval. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 652–655, Washington, DC, USA, 2004. IEEE Computer Society.

[KC92] R. Krovetz and W. B. Croft. Lexical Ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, 10(2):115–141, 1992.

[KC03] D.-H. Kim and C.-W. Chung. QCluster: Relevance Feedback using Adaptive Clustering for Content-based Image Retrieval. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 599–610, New York, NY, USA, 2003. ACM Press.

[KJC04] F. Kang, R. Jin, and J. Y. Chai. Regularizing Translation Models for Better Automatic Image Annotation. In *CIKM '04: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 350–359, New York, NY, USA, 2004. ACM Press.

[KK93] T. Kurita and T. Kato. Learning of Personal Visual Impression for Image Database Systems. In *Proceedings of IEEE International Conference on Document Analysis and Recognition*, pages 547–552, 1993.

[KK01] J. Kahan and M.-R. Koivunen. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM Press.

[KKK03] J.-H. Kang, C.-S. Kim, and E.-J. Ko. An XQuery Engine for Digital Library Systems. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, page 400, Washington, DC, USA, 2003. IEEE Computer Society.

[KP99] E. J. Keogh and M. J. Pazzani. Relevance Feedback Retrieval of Time Series Data. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 183–190, New York, NY, USA, 1999. ACM Press.

[KSR99] S. Kulkarni, B. Srinivasan, and M. V. Ramakrishna. Vector-Space Image Model (VSIM) for Content-Based Retrieval. In *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, page 899, Washington, DC, USA, 1999. IEEE Computer Society.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, V1(4):321–331, 1988.

[Lan68] F. W. Lancaster. *Information Retrieval Systems: Characteristics, Testing and Evaluation.* John Wiley and Sons, New York, NY, USA, 1968.

[LHZ+00] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang. A Unified Framework for Semantics and Feature based Relevance Feedback in Image Retrieval Systems. In *MULTIMEDIA '00: Proceedings of the Eighth ACM International Conference on Multimedia*, pages 31–37, New York, NY, USA, 2000. ACM Press.

[Lie97] H. Lieberman. Autonomous Interface Agents. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 67–74, New York, NY, USA, 1997. ACM Press.

[Lim01] J.-H. Lim. Building Visual Vocabulary for Image Indexation and Query Formulation. In *Pattern Analysis and Applications (Special Issue on Image Indexation)*, volume 4, pages 125–139, 2001.

[LKB+02] M. Lux, W. Klieber, J. Becker, K. Tochtermann, H. Mayer, H. Neuschmied, and W. Haas. XML and MPEG-7 for Interactive Annotation and Retrieval using Semantic Meta-data. *Journal of Universal Computer Science*, 8(10):965–984, 2002.

[LKP95] D. L. Lun, Y. M. Kim, and G. Patel. Efficient Signature File Methods for Text Retrieval. In *IEEE Transactions on Knowledge and Data Engineering*, volume 7, pages 423–435, Piscataway, NJ, USA, 1995. IEEE Educational Activities Department.

[Los] Lost Art Internet Database. `http://www.lostart.de`.

[LS] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C, February 1999.

[LS01] L. Liu and S. Sclaroff. Deformable Shape Detection and Description via Model-based Region Grouping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:475–489, 2001.

[LTM03] J.-H. Lim, Q. Tian, and P. Mulhem. Home Photo Content Modeling for Personalized Event-Based Retrieval. *IEEE MultiMedia*, 10(4):28–37, 2003.

[LVC+99] W.-S. Li, Q. Vu, E. Chang, D. Agrawal, K. Hirata, S. Mukherjea, Y.-L. Wu, C. Bufi, C.-C. K. Chang, Y. Hara, R. Ito, Y. Kimura, K. Shimazu, and Y. Saito. PowerBookmarks: a System for Personalizable Web Information Organization, Sharing, and Management. In *SIGMOD '99: Proceedings of the ACM International Conference on Management of Data*, pages 565–567, New York, NY, USA, 1999. ACM Press.

[McD97] S. McDonald. A Context-based Model of Semantic Similarity. 1997.

[MCR02] H. Moon, R. Chellappa, and A. Rosenfeld. Optimal Edge-based Shape Detection. *IEEE Transactions on Image Processing*, 11(11):1209–1227, 2002.

[Mec95] M. Mechkour. EMIR^2: An Extended Model for Image Representation and Retrieval. In *DEXA'95: Database and EXpert system Applications*, pages 395–404, London, UK, 1995. Springer-Verlag.

[MM90] U. Manber and G. Myers. Suffix Arrays: A New Method for On-Line String Searches. In *SODA '90: Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 319–327, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[MSKP02] J. M. Martínez Sanchez, R. Koenen, and R. Pereira. MPEG-7: The Generic Multimedia Content Description Standard, Part 1. *IEEE MultiMedia*, 9(2):78–87, 2002.

[MTO99] Y. Mori, H. Takahashi, and R. Oka. Image-to-Word Transformation based on Dividing and Vector Quantizing Images with Words. In *MISRM '99: First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[NBE$^+$94] W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC Project: Querying Images by Content using Color, Texture, and Shape. In *Proceedings of Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187. The International Society for Optical Engineering (SPIE), 1994.

[OBCM02] M. Ortega-Binderberger, K. Chakrabarti, and S. Mehrotra. An Approach to Integrating Query Refinement in SQL. In *EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology*, pages 15–33, London, UK, 2002. Springer-Verlag.

[OBM03] M. Orthega-Binderberger and S. Mehrotra. *Relevance Feedback Techniques in Multimedia Databases.* Handbook of Video Databases: Design and Applications, CRC Press, Boca Raton, USA, 2003.

[ORC$^+$98] M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, S. Mehrotra, and T. S. Huang. Supporting Ranked Boolean Similarity Queries in MARS. In *IEEE Transactions on Knowledge and Data Engineering*, volume 10, pages 905–925, Piscataway, NJ, USA, 1998. IEEE Educational Activities Department.

[PAA$^+$87] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. T. H. Romeny, and J. B. Zimmerman. Adaptive Histogram Equalization and Its Variations. In *Computer Vision, Graphics, and Image Processing*, volume 39, pages 355–368, San Diego, CA, USA, 1987. Academic Press Professional, Inc.

[Par88] C. L. Paris. Tailoring Object Descriptions to a User's Level of Expertise. In *Computational Linguistics*, volume 14, pages 64–78, Cambridge, MA, USA, 1988. MIT Press.

[PC98] J. M. Ponte and B. W. Croft. A language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 275–281, New York, NY, USA, 1998. ACM Press.

[Pen03] J. Peng. Multi-Class Relevance Feedback Content-Based Image Retrieval. *Computer Vision and Image Understanding*, 90(1):42–67, 2003.

[PH92] P. Perez and F. Heitz. Multiscale Markov Random Fields and Constrained Relaxation in Low Level Image Analysis. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3:61–64, 1992.

[PL03] C.-M. Pun and M.-C. Lee. Log-Polar Wavelet Energy Signatures for Rotation and Scale Invariant Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):590–603, 2003.

[PM95] R. W. Picard and T. P. Minka. Vision Texture for Annotation. In *Multimedia Systems*, volume 3, pages 3–14, 1995.

[PMO99] K. Porkaew, S. Mehrotra, and M. Ortega. Query Reformulation for Content Based Multimedia Retrieval in MARS. In *ICMCS '99: Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, volume 2, page 747, Washington, DC, USA, 1999. IEEE Computer Society.

[PMS96] R. W. Picard, T. P. Minka, and M. Szummer. Modeling User Subjectivity in Image Libraries. In *IEEE International Conference On Image Processing*, volume 2, pages 777–780, Lausanne, Switzerland, 1996.

[PP00] C. Papageorgiou and P. Poggio. A Trainable System for Object Detection. In *International Journal of Computer Vision*, volume 38, pages 15–33, Hingham, MA, USA, 2000. Kluwer Academic Publishers.

[PPPS03] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos. Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372, 2003.

[PPS99] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. In *International Journal of Computer Vision*, volume 18, pages 233–254, 1999.

[PYDF04] J. Pan, H. Yang, P. Duygulu, and C. Faloutsos. Automatic Image Captioning. In *ICME '04: IEEE International Conference on Multimedia and Expo*, volume 3, pages 1987–1990, 2004.

[RHM97] Y. Rui, T. Huang, and S. Mehrotra. Content-Based Image Retrieval with Relevance Feedback in MARS. In *ICIP '97: Proceedings of the IEEE International Conference On Image Processing*, pages 815–818, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

[RHM98] Y. Rui, T. Huang, and S. Mehrotra. Relevance Feedback Techniques in Interactive Content-Based Image Retrieval. In *Storage and Retrieval for Image and Video Databases*, pages 25–36. The International Society for Optical Engineering (SPIE), 1998.

[RHM99] Y. Rui, T. Huang, and S. Mehrotra. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[Rij79] C. J. van Rijsbergen. *Information Retrieval, 2nd Edition.* Butterworths, London, 1979.

[RJ88] S. E. Robertson and K. S. Jones. Relevance Weighting of Search Terms. pages 143–160, 1988.

[Roc71] J. J. Rocchio. Relevance Feedback in Information Retrieval. In *SMART Retrieval System-Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, 1971.

[Sal68] G. Salton. *Automatic Information Organization and Retrieval.* New York: McGraw-Hill, USA, 1968.

[Sap06] A. D. Sappa. Unsupervised Contour Closure Algorithm for Range Image Edge-based Segmentation. *IEEE Transactions on Image Processing*, 15(2):377–384, 2006.

[SB88] G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. In *Information Processing and Management*, volume 24, pages 513–523. Pergamon Press, Inc., 1988.

[SB90] G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[SB91] M. J. Swain and D. H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SBWR06] B. Shah, R. Benton, Z. Wu, and V. Raghavan. Automatic and Semi-Automatic Techniques for Image Annotation. In *Semantic-Based Visual Information Retrieval*, pages 112–134. IDEA Group Publishing, 2006.

[SC96] J. R. Smith and S.-F. Chang. VisualSEEk: A Fully Automated Content-Based Image Query System. In *Proceedings of the Fourth International Conference on Multimedia*, pages 87–98, New York, NY, USA, 1996. ACM Press.

[SDI06] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.

[SDWW01] A. Th. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. Ontology-Based Photo Annotation. In *IEEE Intelligent Systems*, volume 16, pages 66–74, Piscataway, NJ, USA, 2001. IEEE Educational Activities Department.

[Sha95] W. M. Jr. Shaw. Term-Relevance Computations and Perfect Retrieval Performance. In *Information Processing and Management*, volume 31, pages 491–498, Tarrytown, NY, USA, 1995. Pergamon Press, Inc.

[SKKP03] D. Shin, D. Kim, H. Kim, and S. Park. An Image Retrieval Technique using Rotationally Invariant Gabor Features and a Localization Method. In *ICME 2003: IEEE International Conference on Multimedia and Expo*, volume 2, pages 701–704, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[SM83] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[SNL02] V. Sridhar, M. A. Nascimento, and X. Li. Region-Based Image Retrieval Using Multiple-Features. In *VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, pages 61–75, London, UK, 2002. Springer-Verlag.

[SO95] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Proceedings of the 7th Symposium on Storage and Retrieval for Image and Video Databases*, volume 2420, pages 381–392. The International Society for Optical Engineering (SPIE), 1995.

[SOCP99] P. Salembier, N. E. O'Connor, P. Correia, and F. Pereira. Hierarchical Visual Description Schemes for Still Images and Video Sequences. In *ICIP '99: Proceedings of the IEEE International Conference on Image Processing*, pages 121–125, 1999.

[SRF87] T. K. Sellis, N. Roussopoulos, and C. Faloutsos. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. In *VLDB '87: Proceedings of the 13th International Conference on Very Large Data Bases*, pages 507–518, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.

[SS98a] M. Sarini and C. Strapparava. Building a User Model for a Museum Exploration and Information-Providing Adaptive System. *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia at the Ninth ACM International Hypertext Conference*, pages 63–68, 1998.

[SS98b] A. Stefani and C. Strapparava. Personalizing Access to Web Sites: The SiteIF Project. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia at the Ninth ACM International Hypertext Conference*, Pittsburgh, USA, 1998.

[SS02] P. Salembier and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[Sto02] K. Stolze. Still Image Extensions in Database Systems - A Product Overview. *Datenbank-Spektrum*, 2(2):40–47, 2002.

[SWS$^+$00] A. W. M. 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.

[TC01] S. Tong and E. Chang. Support Vector Machine Active Learning for Image Retrieval. In *MULTIMEDIA '01: Proceedings of the Ninth ACM International Conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[TC02] D. Tjondronegoro and Y.-P. P. Chen. Content-Based Indexing and Retrieval Using MPEG-7 and X-Query in Video Data Management Systems. *World Wide Web*, 5(3):207–227, 2002.

[TPCR04] J. Torres, A. Parkes, and L. Corte-Real. Region-Based Relevance Feedback in Concept-Based Image Retrieval. In *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisboa, Portugal, 2004.

[VC05a] J. Vompras and S. Conrad. A Semi-automated Framework for Supporting Semantic Image Annotation. In *SemAnnot '05: Proceedings of 5th International Workshop on Knowledge Markup and Semantic Annotation at the 4rd International Semantic Web Conference (ISWC 2005)*, pages 105–110. CEUR Workshop Proceedings, 2005.

[VC05b] J. Vompras and S. Conrad. Generating Semantic Templates to Support the Image Annotation Process. In *EWIMT 2005: The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media*, pages 89–95, London, 2005. IEE The Institution of Electrical Engineers.

[VC06] J. Vompras and S. Conrad. Unifying Different Users' Interpretations and Levels of Abstraction for Improving Annotation-based Image Retrieval. In *SMAP '06: 1st International Workshop on Semantic Media Adaptation and Personalization*, pages 55–61, Washington, DC, USA, 2006. IEEE Computer Society.

[VC08a] J. Vompras and S. Conrad. Enhancing Semantic Image Retrieval by Query Expansion using User-Specific Annotation Profiles. In *EuroIMSA 2008: IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 202–208. Acta Press, 2008.

[VC08b] J. Vompras and S. Conrad. Management and processing of personalized annotations in image retrieval systems. In *Advances in Semantic Media Adaptation and Personalization*, pages 137–155. Springer-Verlag, 2008.

[VSC08] J. Vompras, T. Scholz, and S. Conrad. Extracting Contextual Information from Multiuser Systems for Improving Annotation-based Retrieval of Image Data. In *MIR '08: Proceeding of the 1st ACM International Conference on Multimedia Information Retrieval*, pages 149–155, New York, NY, USA, 2008. ACM Press.

[WDS+01] L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. Semi-Automatic Image Annotation. In *INTERACT '01: Proceedings of International Conference on Human-Computer Interaction*, pages 326–333, 2001.

[Wil92] W. J. Wilbur. A Retrieval System Based on Automatic Relevance Weighting of Search Terms. In *ASIS '92: Proceedings of the 55th Annual Meeting on Celebrating Change: Information Management on the*

Move, pages 216–220, Silver Springs, MD, USA, 1992. American Society for Information Science.

[WLWK06] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Probabilistic Document-Context Based Relevance Feedback with Limited Relevance Judgments. In *CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 854–855, New York, NY, USA, 2006. ACM Press.

[WZ02] Y. Wu and A. Zhang. A Feature Re-Weighting Approach for Relevance Feedback in Image Retrieval. *Proceedings of the IEEE International Conference on Image Processing*, pages 581–584, 2002.

[YCWM03] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving Pseudo-Relevance Feedback in Web Information Retrieval using Web Page Segmentation. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 11–18, New York, NY, USA, 2003. ACM Press.

[YHJ03] R. Yan, A. G. Hauptmann, and R. Jin. Negative Pseudo-Relevance Feedback in Content-based Video Retrieval. In *MULTIMEDIA '03: Proceedings of the Eleventh ACM International Conference on Multimedia*, pages 343–346, New York, NY, USA, 2003. ACM Press.

[You] Youtube. `http://www.youtube.com/`.

[ZG02] R. Zhao and W. I. Grosky. Bridging the Semantic Gap in Image Retrieval. In *Distributed Multimedia Databases: Techniques & Applications*, pages 14–36, Hershey, PA, USA, 2002. Idea Group Publishing.

[ZH03] X. S. Zhou and T. S. Huang. Relevance Feedback in Image Retrieval: A Comprehensive Review. In *ACM Multimedia Systems Journal: Special Issue on Content-based Image Retrieval*, volume 8, pages 536–544, Berlin/Heidelberg, 2003. Springer-Verlag.

[ZH06] J. Zhang and S.-W. Ha. A Novel Approach Using Edge Detection Information for Texture Based Image Retrieval. In *ICNC 2006: Proceedings of the Second International Conference on Advances in Natural Computation, Part II*, pages 797–800, 2006.

[ZZ00] L. Zhu and A. Zhang. Supporting Multi-Example Image Queries in Image Databases. In *ICME 2000: IEEE International Conference on Multimedia and Expo (II)*, volume 2, pages 697–700, New York, NY, USA, 2000. IEEE Computer Society.

List of Figures

List of Tables