

**Marcin Detyniecki Peter Knees
Andreas Nürnberger Markus Schedl
Sebastian Stober (Eds.)**

LNCS 6817

Adaptive Multimedia Retrieval

Context, Exploration, and Fusion

**8th International Workshop, AMR 2010
Linz, Austria, August 2010
Revised Selected Papers**

 **Springer**

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Marcin Detyniecki Peter Knees
Andreas Nürnberger Markus Schedl
Sebastian Stober (Eds.)

Adaptive Multimedia Retrieval

Context, Exploration, and Fusion

8th International Workshop, AMR 2010
Linz, Austria, August 17-18, 2010
Revised Selected Papers

Volume Editors

Marcin Detyniecki

Université Pierre et Marie Curie, CNRS research associate, LIP6
4 place Jussieu, 75005 Paris, France
E-mail: marcin.detyniecki@lip6.fr

Peter Knees

Johannes Kepler University, Department of Computational Perception
Altenberger Straße 69, 4040 Linz, Austria
E-mail: peter.knees@jku.at

Andreas Nürnberger

Otto-von-Guericke University, Faculty of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany
E-mail: andreas.nuernberger@ovgu.de

Markus Schedl

Johannes Kepler University, Department of Computational Perception
Altenberger Straße 69, 4040 Linz, Austria
E-mail: markus.schedl@jku.at

Sebastian Stober

Otto-von-Guericke University, Faculty of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany
E-mail: sebastian.stober@ovgu.de

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-27168-7

e-ISBN 978-3-642-27169-4

DOI 10.1007/978-3-642-27169-4

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011942916

CR Subject Classification (1998): H.4, H.3, I.4, H.5.1, H.5, I.2, H.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book contains a selection of the revised contributions that were initially submitted to the International Workshop on Adaptive Multimedia Retrieval (AMR 2010). The workshop was organized by the Johannes Kepler University in Linz, Austria, during August 17–18, 2010.

Since its foundation, the main goals of the AMR workshop series have been to *provide fresh perspectives on current research activities* and to *intensify the exchange of ideas* between the diverse scientific communities involved in adaptive multimedia retrieval, such as multimedia research, human-computer interaction, user modeling, personalization, and machine learning and artificial intelligence.

In this spirit, the first three events were co-located with artificial intelligence-related conferences: in 2003 as a workshop of the German Conference on Artificial Intelligence (KI), in the following year as part of the European Conference on Artificial Intelligence (ECAI 2004) and in 2005 co-located to the International Joint Conference on Artificial Intelligence (IJCAI). Because of its success, in 2006 the University of Geneva, Switzerland, organized the workshop for the first time as a stand-alone event; and since then it has been so: AMR 2007 was organized by the Laboratoire d'Informatique de Paris VI (LIP6) in France and AMR 2008 by the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute (HHI) in Berlin. Its 2009 edition was hosted by the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.

In the revised contributions of the 2010 workshop contained in this edition, the authors present a multitude of novel ideas around three main topics: *context*, *exploration*, and *fusion*. These trends can be observed to play an important role for various types of media, as diverse as text, images, music, and videos. In the contributions utilizing contextual information, this kind of information is predominantly used for improving personalization of search. More precisely, semantic ontologies and cross-digital library retrieval represent two upcoming topics covered. Exploration and discovery of various types of multimedia content is tackled by numerous contributions in manifold manners. They include innovative approaches to video retrieval, music research and retrieval, adaptive similarity measurement, finding, exploring, and structuring multimedia content. Therefore, the employed techniques span a wide field from semantic indexing, to content-based description, to novel ideas in query formulation. Last but not least, media fusion appears to be another emergent research trend, as witnessed by several contributions. Taking into account the user's manifold and context-dependent information needs, integration of different sources and/or results are promising strategies to alleviate limitations of unimodal approaches.

Not least due to the research focus of the organizing institution, research on adaptive music retrieval played a major role in AMR 2010. This is also reflected by an invited contribution referring to one of the workshop's keynote speeches.

Important challenges addressed by the respective contributions include elaborating serendipitous music artist recommenders as well as comprehensive investigations of the potential of similarity measures to model musical similarity as perceived by humans.

We believe that the above trends are representative and thus this book provides a good and conclusive overview of the current research in the area of adaptive multimedia retrieval. We would like to thank all members of the Program Committee for supporting us in the reviewing process, the workshop participants for their willingness to revise and extend their papers for this book, the sponsors for their financial help, and Alfred Hofmann from Springer for his support in the publishing process.

March 2011

Marcin Detyniecki
Peter Knees
Andreas Nürnberger
Markus Schedl
Sebastian Stober

Organization

Program Chairs

Marcin Detyniecki	CNRS, Laboratoire d'Informatique de Paris 6, France
Peter Knees Andreas Nürnberger	Johannes Kepler University, Linz, Austria Otto-von-Guericke-University, Magdeburg, Germany
Markus Schedl	Johannes Kepler University, Linz, Austria

Technical Chair

Sebastian Stober	Otto-von-Guericke-University, Magdeburg, Germany
------------------	---

Local Organization

Peter Knees	Johannes Kepler University, Linz, Austria
Markus Schedl	Johannes Kepler University, Linz, Austria

Program Committee

Thomas Bärecke	Université Pierre et Marie Curie, Paris, France
Jenny Benois-Pineau	University of Bordeaux, LABRI, France
Stefano Berretti	Università degli Studi di Firenze, Italy
Susanne Boll	University of Oldenburg, Germany
Eric Bruno	University of Geneva, Switzerland
Juan Cigarrán	Universidad Nacional de Educación a Distancia, Spain
Ana M. García Serrano	Universidad Nacional de Educación a Distancia, Spain
Fabien Gouyon	INESC Porto, Portugal
Xian-Sheng Hua	Microsoft Research, Beijing, China
Philippe Joly	Université Paul Sabatier, Toulouse, France
Gareth Jones	Dublin City University, Ireland
Joemon Jose	University of Glasgow, UK
Stefanos Kollias	National Technical University of Athens, Greece
Stéphane Marchand-Maillet	University of Geneva, Switzerland
Trevor Martin	University of Bristol, UK

José María Martínez Sánchez	Universidad Autónoma de Madrid, Spain
Bernard Merialdo	Institut Eurécom, Sophia Antipolis, France
Nuria Oliver	Telefónica R&D, Spain
Gabriella Pasi	Università degli Studi di Milano-Bicocca, Italy
Stefan Rürger	The Open University, Milton Keynes, UK
Simone Santini	Universidad Autonoma de Madrid, Spain
Raimondo Schettini	Università degli Studi di Milano-Bicocca, Italy
Ingo Schmitt	University of Cottbus, Germany
Alan F. Smeaton	Dublin City University, Ireland
Arjen P. de Vries	CWI, Amsterdam, The Netherlands

Supporting Institutions

Johannes Kepler University (JKU), Linz, Austria
Otto-von-Guericke-University (OvGU), Magdeburg, Germany
Laboratoire d'Informatique de Paris 6 (LIP6), France
Centre National de la Recherche Scientifique (CNRS), France

Table of Contents

Invited Contribution

- Towards a Storytelling Approach for Novel Artist Recommendations 1
Stephan Baumann, Rafael Schirru, and Bernhard Streit

Context-Based Personalization

- A Survey of Context-Aware Cross-Digital Library Personalization 16
Ana Nika, Tiziana Catarci, Yannis Ioannidis, Akrivi Katifori, Georgia Koutrika, Natalia Manola, Andreas Nürnberger, and Manfred Thaller
- An Ontology-Based Approach of Multimedia Information Personalized Search 31
Mihaela Brut and Florence Sedes

Media Information Fusion

- Approaching Multimedia Retrieval from a Polyrepresentative Perspective 46
David Zellhöfer and Ingo Schmitt
- Knowledge Based Multimodal Result Fusion for Distributed and Heterogeneous Multimedia Environments: Concept and Ideas 61
Florian Stegmaier, Tobias Bürger, Mario Döller, and Harald Kosch

Video Retrieval

- A Contour-Color-Action Approach to Automatic Classification of Several Common Video Genres 74
Bogdan E. Ionescu, Christoph Rasche, Constantin Vertan, and Patrick Lambert
- Differences in Video Search Behavior between Novices and Archivists 89
Henning Rode, Theodora Tsirikika, and Arjen P. de Vries
- An Affect-Based Video Retrieval System with Open Vocabulary Querying 103
Ching Hau Chan and Gareth J.F. Jones

Audio and Music Retrieval

A Comparison of Human, Automatic and Collaborative Music Genre Classification and User Centric Evaluation of Genre Classification Systems 118
Klaus Seyerlehner, Gerhard Widmer, and Peter Knees

Clubmixer: A Presentation Platform for MIR Projects 132
Alexander Schindler and Andreas Rauber

Adaptive Similarities

Similarity Adaptation in an Exploratory Retrieval Scenario 144
Sebastian Stober and Andreas Nürnberger

Similarity Query Postprocessing by Ranking 159
Petra Budikova, Michal Batko, and Pavel Zezula

Finding and Organizing

Proximity-Based Order-Respecting Intersection for Searching in Image Databases 174
Tomas Homola, Vlastislav Dohnal, and Pavel Zezula

Experiences with Shape Classification through Fuzzy *c*-Means Using Geometrical and Moments Descriptors 189
Ugo Erra and Sabrina Senatore

Quantum Logic Based MPEG Query Format Algebra 204
Mario Döllner, Sebastian Lehrack, Harald Kosch, and Ingo Schmitt

Author Index 221

Towards a Storytelling Approach for Novel Artist Recommendations

Stephan Baumann, Rafael Schirru, and Bernhard Streit

German Research Center for Artificial Intelligence,
Alt-Moabit 91c,
10559 Berlin, Germany
{baumann,schirru}@dfki.de, info@bstreit.de

Abstract. The Semantic Web offers huge amounts of structured and linked data about various different kinds of resources. We propose to use this data for music recommender systems following a storytelling approach. Beyond similarity of audio content and user preference profiles, recommender systems based on Semantic Web data offer opportunities to detect similarities between artists based on their biographies, musical activities, etc. In this paper we present an approach determining similar artists based on freely available metadata from the Semantic Web. An evaluation experiment has shown that our approach leads to more high quality novel artist recommendations than well-known systems such as Last.fm and Echo Nest. However the overall recommendation accuracy leaves room for further improvement.

Keywords: Music Recommender Systems, Artist Similarities, Semantic Web, Linked Data, Storytelling Approach.

1 Introduction

With the advent of the Semantic Web large amounts of structured and interconnected data about various different kinds of resources have become freely available. Our vision is to exploit this data for music recommender systems that tell a story about a recommended song.

Let's start with an illustrating example: The meanwhile popular bands *Air* and *Phoenix* started their international careers by focusing on their French roots. They grew up in *Versailles* and *Paris* and established several relations to befriended artists and band members of, e. g., *ORANGE* and *Cassius* in the early *nineties*. *Etienne de Crécy* was the tour manager of *ORANGE* and worked later on together with *Philippe Zdar* of *Cassius*. After being highly influential for the creation of the so-called *French house* music genre in the *mid-nineties* *Cassius* even cooperated at the later stage of their career with *Pharell Williams* who is an American recording artist and producer of *pop*, *hip hop*, and *R+B* music. He performed as a guest for their song *Eye Water* in *2007*.

The musical facts behind these small stories are already freely accessible and represented in standard Semantic Web formats. Our core idea is to compute

recommendations which are based on such data from the Semantic Web to go beyond the comparison of audio signals or user preference profiles as it is done in traditional content-based or collaborative filtering systems. At the writing of this paper we can already state that the richness and topicality of this data is very promising to elaborate fine-grained relationships. They can include an artist’s biography or her musical activities comprising the place where the artist was born, the record label she works for, musical cooperations, and so on.

In the paper we propose a first approach towards this vision. We collected structured data about the genres artists are associated with as well as the release years of their records. We aggregate this data thus obtaining descriptions that can be used to determine similarities between artists. A twofold evaluation study has been conducted. In the first step we checked how close the artist similarities determined by our approach are to those provided by Last.fm¹. Afterwards, in the second step we evaluated the novelty and perceived quality of our recommendations in a user study. The evaluation showed that despite the overall recommendation quality cannot keep up with popular systems such as Last.fm and Echo Nest² our approach leads to more high quality novel artist recommendations than those systems.

The remainder of this paper is structured as follows: In Section 2 we present related work in the field of (music) recommender systems based on Semantic Web data and report on our previous work. Section 3 presents the state of the art in music recommender systems and points out current issues. Next in Section 4 we give a short introduction to the nature of the Semantic Web and present data sources that can be used to obtain metadata about artists. Section 5 depicts the details of our proposed approach. In Section 6 we present the results of our evaluation experiments. We conclude the paper in Section 7 by summarizing our findings and presenting ideas for future work.

2 Related Work

In [10] Passant introduces an approach exploiting linked data³ to generate resource recommendations. The method calculates the semantic distance between resources by analyzing their link structure. It takes direct and indirect links into account. The algorithm is among others applied in the domain of music. Here the author uses the DBpedia⁴ data set to generate the recommendations. The algorithm takes a seed URI as input and computes the distance between this URI and all other resources from the data set. To provide relevant recommendations the result is limited to instances of `dbpedia-owl:MusicArtist` and `dbpedia-owl:Band`. An evaluation is planned based on user feedback about the recommended artists and bands.

¹ <http://www.last.fm/>

² <http://the.echonest.com/>

³ <http://linkeddata.org/>

⁴ <http://dbpedia.org/>

Ziegler ([15]) proposes an approach for recommendations in decentralized systems based on Semantic Web data. His method requires a taxonomy from the domain in which recommendations have to be provided (e. g., a domain taxonomy for books). User profiles are generated based on this taxonomy. Whenever a user expresses a preference for an item, the associated categories in the taxonomy receives a positive rating score in the user’s profile. The preference expression can also be inherited to super concepts in the taxonomy. That way taxonomy-based user profiles evolve that should be used in combination with trust networks (direct and indirect trust relationships between users) to determine a user’s nearest neighbors. According to Ziegler, this approach is less vulnerable to manipulation compared to pure rating based nearest neighbors detection. Also it does not require that users rate exactly the same items to become nearest neighbors. Rating items from similar categories is already sufficient.

In [2] we presented an approach extracting cultural metadata about artists from websites thus enabling artist recommendations. Per artist 50 websites containing reviews of their musical work were crawled and the meaningful parts of these sites were extracted. By making use of part-of-speech tagging we extracted features characterizing the artists such as single occurrences of nouns or adjectives. In the next step these features were weighted by applying the TF-IDF measure. The weighted features were then mapped to a vector space. Artist similarities were determined by computing their cosine similarity. The evaluation experiments showed that recommendations based on cultural metadata were able to predict expert as well as end-user perception of similarity.

3 Music Recommender Systems

In the recommender systems literature there are three predominant approaches to generate recommendations ([1]):

Content-based (CB) methods estimate the utility of an item according to its similarity to items for which the user has expressed a preference in the past. To capture the content of a song one widely adapted approach is to divide the song into frames for which a spectral representation is obtained. The mel-frequency cepstral coefficients (MFCCs) are often used for that purpose. That way an audio item can be represented, e. g., as a Gaussian Mixture Model (GMM) ([8]) or as a set of clusters of its associated frames ([9]). To determine the similarity between audio items that are represented that way Liu and Huang propose a parametric distance metric for the GMMs and Logan and Salmon propose to use the earth mover’s distance metric ([12]) for the cluster representation respectively. CB recommender systems suffer the problem of overspecialization, i. e., songs that are potentially relevant for a user are not recommended in case that they are not similar to the user’s previously preferred songs. Also CB methods cannot take the quality of the recommended items into account.

Collaborative Filtering (CF) approaches recommend items that users with similar tastes as the active user have liked in the past (e. g., [11][13]). They capture item preferences either explicitly by requiring the users to rate items

or implicitly by observing the users' browsing behavior. That way a matrix of user ratings for items is obtained on which the pairwise similarity between users can be calculated. Popular similarity measures for CF algorithms are the Pearson correlation coefficient as well as cosine similarity. A major drawback of CF systems is the so called new item problem, i. e., items which are new in a system cannot be recommended unless they obtained a minimum amount of ratings by users. Further CF systems tend to recommend popular items only that way making access to the long-tail of rarely rated items difficult. For users with unusual tastes it is often hard to find peers in the system thus leading to poor quality recommendations.

To overcome the drawbacks of each individual approach and to combine their assets *hybrid systems* are often build. In [3] Burke describes methods to combine different recommendation approaches.

4 Semantic Web Data Sources

Our approach relies on data from the Semantic Web to represent music artists or bands as well as the connections between them. In [7] Berners-Lee et al. describe the Semantic Web as a new form of Web content that is meaningful to computers. It is characterized by five points:

Expressing meaning: The Semantic Web aims to bring a meaningful structure to the content of Web pages, that way creating an environment in which software agents that roam from page to page can carry out sophisticated tasks for users. It is an extension of the current Web, giving well-defined meaning to information in order to improve the cooperation between people and computers.

Knowledge Representation: To make the Semantic Web function, it is necessary that computers have access to structured collections of information as well as sets of inference rules that can be exploited for automated reasoning. A fundamental technology for developing the Semantic Web is the Resource Description Framework (RDF) [5]. RDF encodes meaning in sets of triples (like subject, verb, and object) that can be written using XML tags. Using RDF, a document asserts that certain things, such as people, Web pages, etc., have properties (e. g., "is a sister of") with specific values (e. g., another person).

Ontologies: In Artificial Intelligence and Web research the term ontology refers to a document that defines the relations among terms. Typically it consists of a taxonomy and a set of inference rules. E. g., an address may be modeled as a type of location, city codes may be modeled to apply only to locations, etc.

Agents: The power of the Semantic Web will be realized when programs are implemented that collect Web content from different sources, process the information and exchange it with other programs. Example applications comprise e. g., *Proofs* (verification that a person is the one you were looking for) or *Digital signatures* (automatic verification that some information has been provided by a specific trusted source).

⁵ <http://www.w3.org/RDF/>

Evolution of Knowledge: The Semantic Web enables everybody to express new concepts by just naming them with a URI. These concepts can then be progressively linked into a universal Web that way opening up the knowledge of humans for meaningful analysis by software agents.

Subsequently we will describe two Semantic Web data sources that contain structured metadata about artists.

4.1 Freebase

Freebase⁶ is an online collection of structured data that has been harvested from many different sources. It also includes direct wiki-like contributions provided by the community of users that way forming a large collaborative knowledge base. The aim of Freebase is the creation of a global resource allowing people and machines to access common information in a convenient way. The Freebase data is available under the Creative Commons Attribution 2.5 Generic license⁷. It can be accessed via a dedicated API, an RDF endpoint, as well as database dumps. Freebase is developed by the American software company Metaweb and is publicly available since March 2007.

We currently extract the following metadata from Freebase:

- *origin*: The place (city or country) where an artist or group started their career.
- *instrument*: The instrument(s) an artist plays.
- *genre*: The musical genre of the artist or group.
- *artist collaboration*: Artist collaborations that appear in Freebase. The collaboration is given a name consisting of all artists or groups that participate (e. g. “2pac/and notorious b.i.g/trapp”), and all appearing artists or groups are linked to this name.
- *record release year*: We extract all years in which an artist or group has released a record.

4.2 DBpedia

The DBpedia⁸ project is a community effort that aims at extracting structured information from Wikipedia and making the information available on the Web. That way it covers many domains, represents community agreement, and it automatically evolves as Wikipedia changes. Access to the DBpedia data set is granted online via a SPARQL query endpoint and as Linked Data. Further the data can be downloaded as text files either in N-Triples or in N-Quads format. DBpedia is licensed under the Creative Commons Attribution-Share Alike 3.0⁹ license and the GNU Free Documentation License¹⁰.

⁶ <http://www.freebase.com/>

⁷ <http://creativecommons.org/licenses/by/2.5/>

⁸ <http://dbpedia.org/>

⁹ <http://creativecommons.org/licenses/by-sa/3.0/>

¹⁰ http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License/

Currently we only extract the category labels from DBpedia. These are the labels that are shown on the bottom of most of the Wikipedia pages. For example, for Madonna we have categories “1958 births”, “1980s singers”, “1990s singers”, “American female singers”, etc.

5 Our Approach

To determine similar artists we first select a set of properties (features) of which we think that they best characterize the artists and map them to a vector space. We discard those features that are not discriminative and apply feature weighting on the remaining ones. The pairwise similarity between artists is then calculated by determining their cosine similarity. Figure 1 presents an overview of our proposed approach and the current section will present the single steps in greater detail.

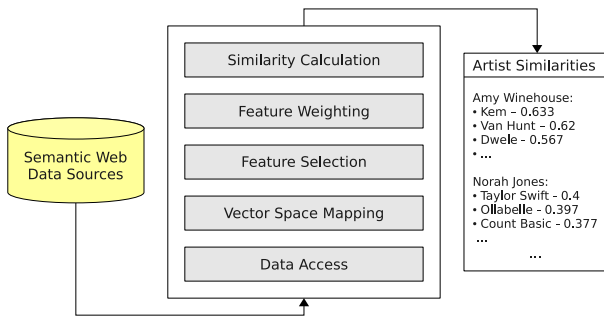


Fig. 1. Process steps to determine artist similarities

5.1 Artist Properties

The Semantic Web data sources provide a variety of features that can be used to describe how artists are related. In the current version of our system we use artist metadata from Freebase. We combine the genres that are annotated for the artists together with their record release years as follows:

1. Get all genres for an artist, e. g., “pop music” and “dance music”.
2. Get all record release years for the artist, e. g., “1973” and “1980”.
3. Convert the record release years into a set of classes that cover the years by overlapping decades, e. g., “1965-1975” and “1970-1980”. We use the short notation “1965s” and “1970s” respectively.
4. Now we merge the properties by forming the product of all property sets. I. e., we multiply the genres “pop music” and “dance music” by the record release decades “1965s” and “1970s” for the artist and obtain the four merged properties “pop music, 1965s”, “pop music, 1970s”, “dance music, 1965s” and “dance music, 1970s”.

We will refer to our approach using merged genres and record release years as *MergedGenres* subsequently.

5.2 Vector Space Mapping

In our data base we have 11,276 artists and 5,020 properties. We map the artist representations to a vector space. Every property in the set of all artist properties constitutes one dimension in the vector space. The feature values are either set to one if the property applies for the artist or to zero otherwise. An exemplary extract of a simple artist-property matrix with artists and their associated genres is depicted in Figure 2.

	Pop Music	Dance-pop	Electronica	Contemp. R&B	⋮
Madonna	1	1	1	0	
Britney Spears	1	1	0	1	⋮
Lady Gaga	1	0	1	1	
...	...				

Fig. 2. Exemplary extract of an artist-property matrix for artists and their associated genres

5.3 Feature Selection

Very rare and very frequent features are not likely to discriminate artists appropriately. For that purpose we remove dimensions representing properties that are annotated less than two times for all artists. When examining the data set in greater detail, we found that there are no features that appear overly often (e. g., with more than 50% of all artists). Therefore it was not necessary to implement an upper bound for the appearance of properties. After the feature selection step the vector space matrix is reduced to 3,560 dimensions.

5.4 Feature Weighting

Artist properties that appear rather rarely are more discriminative than such properties that appear very often. For that reason such features should obtain a higher weight. In information retrieval the term frequency/inverse document frequency (TF-IDF) measure is widely used to achieve this goal ([6]).

Let N be the number of all artists in the data set and k_i a property that is annotated for n_i of them. Further $f_{i,j}$ is the number of times k_i is annotated for i_j .¹¹ The term frequency is computed as follows:

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \quad (1)$$

with the maximum computed over the frequencies $f_{z,j}$ of all properties which are annotated for item i_j . Properties that are annotated for many artists are not discriminative. The measure of the inverse document frequency

$$IDF_i = \log \frac{N}{n_i} \quad (2)$$

is used to cope with this problem. The combined TF-IDF weight for a property is determined by multiplying its term frequency (TF) with its inverse document frequency (IDF):

$$w_{i,k} = TF_{i,j} \times IDF_i \quad (3)$$

5.5 Similarities Calculation

We calculate the pairwise cosine similarity between all artists. Let $ItemProfile(i_1) = (w_{11}, w_{12}, \dots, w_{1n})$ be the vector of properties of item i_1 and $ItemProfile(i_2) = (w_{21}, w_{22}, \dots, w_{2n})$ the vector of properties of item i_2 respectively. Then their cosine similarity is determined as follows:

$$sim(i_1, i_2) = \frac{\vec{i}_1 \cdot \vec{i}_2}{\|\vec{i}_1\|_2 \times \|\vec{i}_2\|_2} = \frac{\sum_{j=1}^n w_{1j} \cdot w_{2j}}{\sqrt{\sum_{j=1}^n w_{1j}^2} \cdot \sqrt{\sum_{j=1}^n w_{2j}^2}} \quad (4)$$

6 Evaluation

The evaluation of our approach is split in two parts: First, we evaluated how close the artist similarities determined by our approach are to those provided by Last.fm. Second, we performed a user experiment with students and employees of the University of Popular Music and Music Business¹² in Mannheim, Germany. The students judged the perceived quality and novelty of artist recommendations provided by our approach.

A combination of objective and subjective evaluation methods for music recommender systems has previously been proposed by Celma and Herrera. In [4] they describe an item-centric evaluation method trying to detect whether the typology of the item-based recommendation network has an intrinsic pathology that inhibits novel recommendations. Further, they measure in a user-centric evaluation the perceived quality of the recommendations.

¹¹ Please note that in our case $f_{i,j} \in \{0, 1\}$ as a property is either set for an artist or not.

¹² <http://www.popakademie.de/>

6.1 Comparison to Last.fm

In the first step of our evaluation experiment we compare the artist similarities provided by Last.fm to those generated by our approach. Last.fm is a service that enables the discovery of music based on the music a user has heard before. Similar artists in Last.fm are calculated on the basis of the listening habits of the users. If many users listen to artist X and also to artists Y and Z then these artists are marked as similar. We consider these artist similarities as a sound ground truth data set as the services claims to harness collective intelligence from more than 40 million active users in more than 200 countries ([14]).

Out of 11,276 artists in our data base our approach was able to determine similar artists for 5,119 of them. For 5,048 of these artists also similar artists from Last.fm were available. The similarity values of both approaches were normalized to be in the range between 0 and 1. To measure how close our artist similarities are to those of the Last.fm reference data set, we use predictive accuracy metrics (e. g., [5], pp. 20-21). In particular the measures *mean absolute error* (MAE) and *root mean squared error* (RMSE) are used. MAE determines the average absolute deviation between the predicted similarity values and the similarity values in the ground truth data set. Let A be a set of N artist pairs ($A = \{a_1, a_2, \dots, a_N\}$), let p_i be the predicted similarity value for artist pair i and let r_i be the similarity value for the artist pair in the ground truth data set respectively. MAE (E_{MAE}) is then calculated as follows:

$$E_{MAE} = \frac{\sum_{i=1}^N |p_i - r_i|}{N} \quad (5)$$

RMSE (E_{RMSE}) puts more emphasis on large errors and is determined as follows:

$$E_{RMSE} = \sqrt{\frac{\sum_{i=1}^N (p_i - r_i)^2}{N}} \quad (6)$$

Our approach was able to determine similarities for 26,168,328 pairs of artists. 2,095,469 of them have a similarity value bigger than 0.1. These are considered subsequently. For 370,370 of these pairs we could obtain similarity values from Last.fm so these pairs are used for the evaluation. Our *MergedGenres* approach reaches a MAE of 0.17 and a RMSE of 0.23 respectively. Table 6.1 presents a detailed overview of the errors in tenth part intervals. With these numbers we were pretty confident that our system could generate reasonable artist recommendations so that we performed a subjective evaluation study in the next step.

6.2 User Experiment

We performed an evaluation experiment with eight students and two employees of the University of Popular Music and Music Business in Mannheim, Germany ($N = 10$). For three of their five most favorite artists we presented them artist

Table 1. Overview of errors of our *MergedGenres* approach in tenth part intervals

Interval	#Occurrences	Percentage
$e < 0.1$	143,576	39%
$0.1 \leq e < 0.2$	105,979	29%
$0.2 \leq e < 0.3$	59,145	16%
$0.3 \leq e < 0.4$	30,465	8%
$0.4 \leq e < 0.5$	15,301	4%
$0.5 \leq e < 0.6$	7,887	2%
$0.6 \leq e < 0.7$	4,142	1%
$0.7 \leq e < 0.8$	2,237	1%
$0.8 \leq e < 0.9$	1,003	0%
$0.9 \leq e$	635	0%

recommendations from Last.fm, our *MergedGenres* approach, and Echo Nest. Echo Nest is a music application platform that can be used to add intelligence to music related applications. The Echo Nest system is powered by the first machine learning platform for music and is based on 12 years of research and development at MIT, Columbia and Berkeley. It performs three kinds of analyzes:

1. It analyzes the content of music according to features such as key, tempo, rhythm, and timbre.
2. Echo Nest analyzes what is written on the web about artists, albums, and songs.
3. It identifies trends by analyzing social networks, p2p systems, blogs, online forums, playlists, etc.

The process of the user experiment was as follows: First the participants were asked to tell us their five favorite musical artists. For every participant we chose three of their favorite artists for which our algorithm was able to determine similar artists. Then we composed a list of 15 recommendations for each favorite artist with five similar artists from each approach. For three participants of our test group, only for two of their five favorite artists *MergedGenre* recommendations could be provided. In this case the *MergedGenre* recommendations were left out and the Last.fm and Echo Nest recommendations were evaluated nevertheless. The participants were not told which recommendation was generated by which system. Then the participants had to judge (1) whether a recommended artist was known to them and (2) how they perceived the quality of the recommendation on a five point rating scale (very good, good, neutral, bad, or very bad).

Figure 3 shows how the participants of the evaluation experiment rated the quality of the recommendations from the three systems. Last.fm and Echo Nest clearly provided more very good and good artist recommendations than our approach. In Figure 4 we aggregated very good and good ratings as positive ratings and neutral, poor, and very poor ratings as negative ratings. 75.33% of the Last.fm recommendations were rated as positive, 51.49% of our *MergedGenres* approach, and 69.59% of the Echo Nest recommendations were rated as positive respectively.

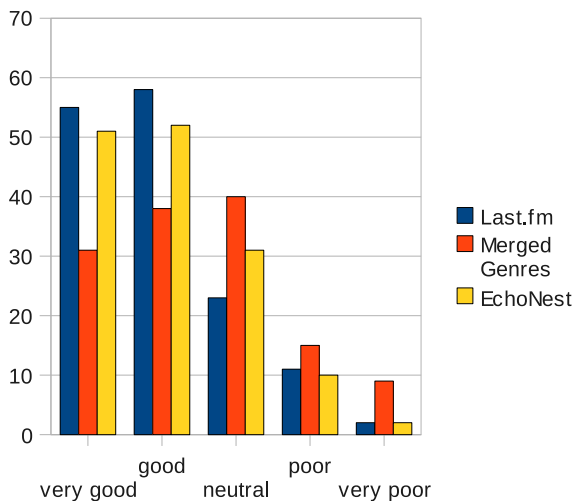


Fig. 3. Assessment of the recommendation quality. The participants of the evaluation study rated the quality of the artist recommendations for three of their five favorite artists.

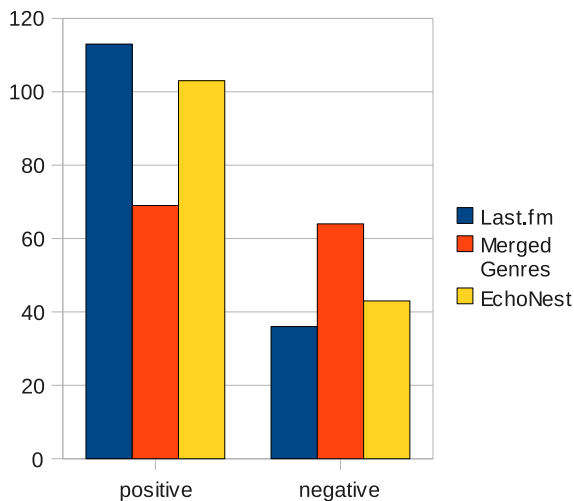


Fig. 4. Aggregated recommendation quality. Very good and good ratings have been aggregated and are juxtaposed to the aggregated neutral, poor, and very poor ratings.

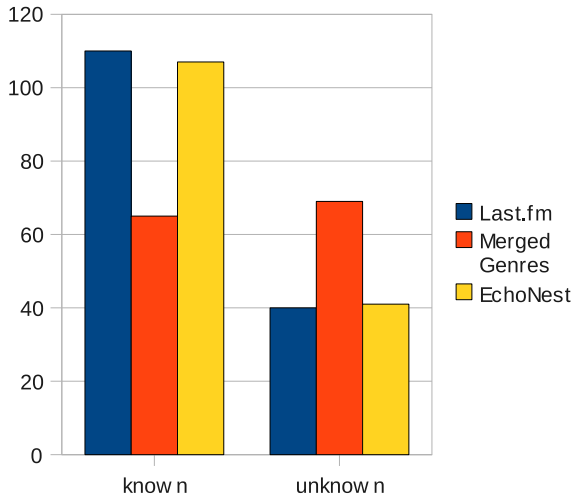


Fig. 5. Evaluation of the recommendation novelty. For every recommended artist the participants of the evaluation experiment said whether they already knew the artist or not.

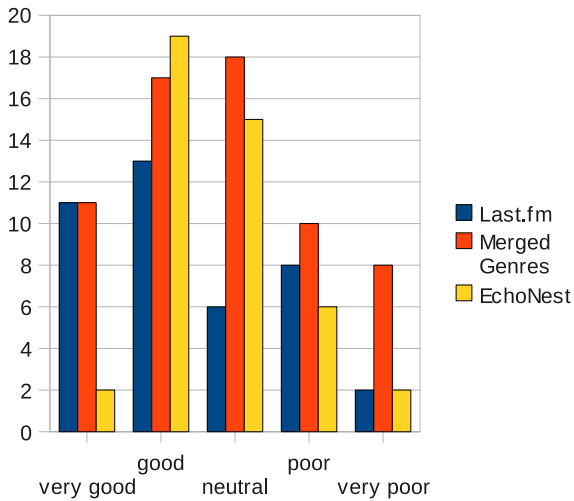


Fig. 6. Evaluation of the quality of novel recommendations. The figure shows the quality ratings for the recommended unknown artists.

However the goal of our approach was to provide recommendations with a high degree of novelty. Figure 5 shows how many unknown artists were recommended by each system. Here our *MergedGenres* approach clearly generates the most novel recommendations. This leads us to further investigate how many of the unknown artists were perceived as good recommendations by the participants of the evaluation experiment.

In Figure 6 we depict the perceived quality of the unknown artists recommendations. If we again aggregate very good, and good ratings as positive ratings and neutral, poor, and very poor ratings as negative ratings (Figure 7) we find that our approach provides the most positive novel artist recommendations. However quality remains an issue for *MergedGenres* as it also generates the most negative novel artist recommendations. The ratios of positive novel recommendations to novel recommendations are 60% for Last.fm, 40.58% for *MergedGenres*, and 51,22% for Echo Nest respectively.

6.3 Discussion

The evaluation experiment has shown that music recommendations based on metadata from the Semantic Web can lead to more high quality novel artist recommendations than widely used systems such as Last.fm and Echo Nest. Yet, the *MergedGenres* approach suffers an issue that is typical for pure content-based recommender systems, namely it is not capable of taking the quality of recommended artists into account. Thus it cannot keep up with the overall recommendation quality of Last.fm and Echo Nest. Collaborative filtering systems

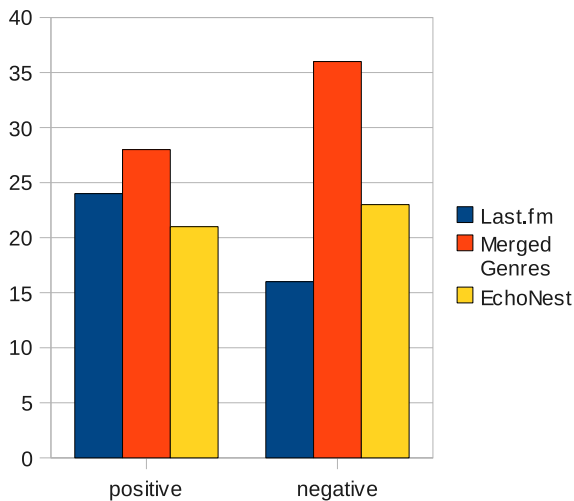


Fig. 7. Aggregated view of the ratings for the novel artists recommendations. Very good and good ratings have been aggregated and are juxtaposed to the aggregated neutral, poor, and very poor ratings.

generally do not suffer from the problem of recommending many poor quality artists as the recommendation algorithm is directly based on the users' quality assessments for artists. However collaborative information filters tend to recommend popular items predominantly (see also [4]) thus making access to the long-tail of less popular artists difficult.

Whether novelty is a desired feature depends to a large amount on the application domain and user preferences. E.g., for personalized radio stations it might be enough to play songs a user likes independent of their novelty. Platforms that sell music on the other hand might profit from recommending novel artists matching the users' personal preferences.

7 Conclusion and Future Work

In this paper we presented an approach for artist recommendations based on structured metadata from the Semantic Web. Our superior goal is to follow a storytelling approach that goes beyond the pure comparison of sonic features and user preference profiles by taking the artists' biographies, musical activities, etc. into account. In the first version of our system we combine the genres and record release years from the Freebase data set as descriptive features for artists. An evaluation experiment has shown that this rather simple approach leads to more high quality novel artist recommendations than popular systems such as Last.fm and Echo Nest.

In our future work we aim at improving the coverage of our approach. With *MergedGenres* we were only able to provide artist recommendations for 32 of 50 named favorite artists. Further the overall recommendation quality of our system needs to be improved. For that purpose the giant graph of linked data should be further exploited to find new and interesting connections between artists.

Acknowledgments. This research has been financed by the IBB Berlin in the project "Social Media Miner", and co-financed by the EFRE funds of the European Union.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 734–749 (2005)
2. Baumann, S., Hummel, O.: Using cultural metadata for artist recommendations. In: *Proceedings of the Conference on Web Delivering of Music (Wedelmusic 2003)*, Leeds, UK (September 2003)
3. Burke, R.D.: Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.* 12(4), 331–370 (2002)
4. Celma, O., Herrera, P.: A new approach to evaluating novel recommendations. In: *RecSys 2008: Proceedings of the 2008 ACM Conference on Recommender Systems*, pp. 179–186. ACM, New York (2008)

5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22(1), 5–53 (2004)
6. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28(1), 11–21 (1972)
7. Lee, B.T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* (May 2001)
8. Liu, Z., Huang, Q.: Content-based indexing and retrieval-by-example in audio. In: *IEEE International Conference on Multimedia and Expo. (II)*, pp. 877–880 (2000)
9. Logan, B., Salomon, A.: A music similarity function based on signal analysis. In: *ICME. IEEE Computer Society, Los Alamitos* (2001)
10. Passant, A.: Measuring semantic distance on linking data and using it for resources recommendations. In: *Proceedings of the AAAI Spring Symposium "Linked Data Meets Artificial Intelligence"* (2010)
11. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: *CSCW 1994: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp. 175–186. *ACM, New York* (1994)
12. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vision* 40(2), 99–121 (2000)
13. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating “word of mouth”. In: *CHI 1995: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 210–217. *ACM Press/Addison-Wesley Publishing Co., New York, NY* (1995)
14. Wikipedia. Last.fm — Wikipedia, the free encyclopedia (2010) (Online accessed June 11, 2010)
15. Ziegler, C.-N.: Semantic Web Recommender Systems. In: Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A. (eds.) *EDBT 2004. LNCS*, vol. 3268, pp. 78–89. *Springer, Heidelberg* (2004)

A Survey of Context-Aware Cross-Digital Library Personalization

Ana Nika¹, Tiziana Catarci², Yannis Ioannidis¹, Akrivi Katifori¹,
Georgia Koutrika³, Natalia Manola¹, Andreas Nürnberger⁴,
and Manfred Thaller⁵

¹ University of Athens, Athens, Greece

{a.nika,yannis,vivi,natalia}@di.uoa.gr

² Sapienza University of Rome, Rome, Italy

catarci@dis.uniroma1.it

³ Stanford University, California, USA

koutrika@stanford.edu

⁴ Otto-von-Guericke University Magdeburg, Magdeburg, Germany

andreas.nuernberger@ovgu.de

⁵ University of Cologne, Cologne, Germany

manfred.thaller@uni-koeln.de

Abstract. The constant interaction of users with different Digital Libraries (DLs) and the subsequent scattering of user information across them raise the need not only for Digital Library interoperability but also for cross-Digital Library personalization. The latter calls for sharing and combining of user-information across different DL systems so that a DL system may take advantage of data from others. To achieve this goal, DL systems should be able to maintain compliant and interoperable user models and profiles that enable propagation and reconciliation of user information across different DLs. In this paper, we motivate the need for cross-Digital Library personalization, we define and examine user model, profile, and context interoperability, and we survey and discuss existing user model interoperability approaches.

Keywords: cross-Digital Library personalization, user model interoperability, user profile, user context, interoperability approaches.

1 Introduction

Whether digitized or born digital, the information found in Digital Libraries (DLs) is growing at an unprecedented rate making it difficult for individuals to identify relevant items in a reasonable amount of time. Furthermore, the new generation of DLs is more heterogeneous than before regarding content diversity and user-community variety. Hence, DLs increasingly need to be more effective at providing information that is tailored to a person's preferences, interests, knowledge, skills, etc. For such personalization to be successful and result in different system behaviors to different users, a DL system needs to provide adequate representation of a user, conforming to a proper *user model* supported

by the system. Each individual instantiation of the user model is called a *user profile*. It is apparent that the profile of a user is also influenced by the *user context*, which is all information that characterizes the environment of a user, resulting in differences in preferences and actions during different interactions of the same user with a DL. Hence, *context models* are necessary as well.

Nowadays, users interact with different DL systems on a regular basis and update their profiles stored at these systems. These distributed and heterogeneous profiles constitute a valuable source of information for DL systems to “understand” their users better and improve their personalization and adaptation services. This user information, however, is not easily transferable from one system to another. To achieve cross-Digital Library personalization, DL systems should go well beyond the traditional techniques for their interoperability (or any other Information System, for that matter), which is usually confined to sharing and mapping of primary content and metadata; they should take into account the precise nature of user profiles and proceed with a new theory for handling “user interoperability”. In addition, they should prevent users from entering the same information into every system, but reuse each other’s profiles freely.

The goal of achieving cross-Digital Library personalization in a collaborative and interoperable setting raises several issues related to different user models that DLs may use, different user profile characteristics captured in different DLs, or even different aspects of the user context. The following example helps to illustrate the above. Consider two different DLs that aim to interoperate. DL A is a historical DL that typically contains multimedia documents illustrating the history of European countries. Each multimedia document is composed of a text describing the history of a country, a video showing important monuments, and several audio files with each country’s traditional music. DL B is a research DL, owned by a research institution, and stores advanced research results. Each DL has its own set of users (user profiles), which may nevertheless, overlap with the other set. The two DLs should communicate in such a way that their users can access them as if they were a single DL. To achieve this goal, the systems’ user models should be interoperable, which is a major challenge: DL A uses a simplified user model, capturing basic characteristics of a user, without providing significant personalization capabilities; on the contrary, DL B supports an enriched user model, records several user preferences, and offers advanced personalization techniques. How can a user of one DL be transferred to the other? Is there a natural mapping between the two different user models? Is it possible for user characteristics captured in both DLs to be consolidated so that a user can have a personalized usage experience in both systems? Are there any privacy issues?

This survey paper describes relevant state-of-the-art approaches that attempt to address these questions. Approaches for user model, profile, and context interoperability have been developed in recent years for general adaptive web-based systems such as recommender and educational systems. Nevertheless, no much effort has been devoted to the creation of relevant solutions for DLs. As cross-Digital Library personalization evolves in a matter of major importance

for future interoperable DLs, it influences its underlying requirements such as user model, profile, and context interoperability to gain the relevant attention. For this reason, the six approaches that will be described in following sections, which were not explicitly developed for DL systems, may have the dynamics to be regarded as suitable for resolving DL-related user interoperability issues. The remainder of this paper is structured in the following way. Section 2 defines user modeling, user profiling, and user context as well as the concept of interoperability for each one, at both the semantic and syntactic levels and, finally, introduces some privacy concerns. Section 3 presents the state-of-the-art approaches related to user model interoperability and then elaborates further on those that are applicable to user profile and user context interoperability. Section 4 discusses the advantages and disadvantages of the approaches identified. Finally, Section 5 concludes the paper and introduces some directions for future research.

2 User Interoperability

As it was mentioned in the previous section, adequate user modeling is important for personalization. Respectively, appropriate *user model interoperability* is essential for cross-Digital Library personalization.

We can define user modeling as the process of capturing all the fundamental information about DL users in order for the system to be able to provide personalized information to different users. We can distinguish user modeling from user profiling by defining user profiling as the process of collecting information about a user in order to generate the user's profile, depending on the current user model. In general, a user model should be rich enough to allow different access to the content and the functionalities provided by the system, to maintain the explicit or implicit preferences affecting the results of the user operations, and to differentiate based on the context of the user. Attributes of the user that may be reflected in a DL are user credentials, demographics, access rights, preferences, interests, background, level of maturity and expertise, etc. Up to now, however, there is no generally accepted user model that may be used in every Digital Library application and ensure that a profile created within a certain DL may be moved effortlessly to another. Thus, interoperability in terms of user modeling refers to the ability of DL systems to support compliant and interoperable user models that enable the propagation of user information across different DLs.

Having a common model or a way to move a user profile from one DL to another is not enough. On one hand, there is the issue of user rights and how they are propagated from one DL to the other. On the other, there is the issue of reconciliation of different and, in some cases, even conflicting preferences or user profile characteristics. It becomes apparent, that another type of interoperability is also needed. Thus, interoperability in terms of user profiling refers to the ability of DL systems to support mechanisms of reconciliation of user profile characteristics.

Moreover, there are "external" factors to the user model related to the context of a user that may affect the profile and result in differences in preferences

and actions when a user interacts with a DL. The issue of user context is an issue that has not been fully explored and defined but is very much related to interoperability and user modeling. Context may include the user “situation”, position, time, role, company of other users, etc. Thus, identifying the relation between user model and user context, as well as revealing where the user model ends and the context begins form important issues for further investigation. Interoperability in terms of user context refers to the ability of DL systems to support compliant context descriptions and interpret user information in a concrete way given the same context.

The interoperability of user models, profiles, and context should be achieved in the *syntactic* and *semantic* level. Syntactic interoperability refers to the capability of different DL systems to interpret the syntax of the delivered user model/profile/context in the same way. Semantic interoperability is concerned with ensuring that the precise meaning of exchanged user model/profile/context characteristics is understandable by any other system.

Finally, privacy issues are very critical for cross-DL personalization [38]. In the context of cross-DL personalization, that requires user models to be shared across systems, privacy is not only related to acquisition of permission from users to collect and use their data, but it is also related to obtainment of users’ explicit consent before transferring user information to other systems. Privacy concerns associated with the latter include a) the systems that can access user data, b) the part of the user profile that can be made available to other systems, c) the time period the user data are retained, etc.

In the following section, a number of approaches that resolve the identified interoperability issues are described unveiling also the realization of syntactic and semantic interoperability.

3 User Model Interoperability Approaches

General User Modeling Systems (GUMS) proposed by Kass and Finnin [19] offer various user modeling services. However, they have seen limited use [20]. Apart from these user modeling approaches, there are also some standardization efforts of user model relevant aspects. The IEEE Public and Private Information (PAPI) specification [10] was created to represent student records. The IMS Learner Information Package (LIP) specification [11] provides a model that represents user attributes required for recording and handling learning history, goals, and achievements. IMS and PAPI are generic and well known standards, but suffer from some disadvantages. They are not conceptually extensible and do not represent dynamic user attributes, like preferences and interests. The work of Orwant called “Doppelgänger” [30] focused mainly on the collection and distribution of user information and the use of several learning methods. In this approach, the basic problem of sharing user data had already been identified.

Recent advances in user model interoperability reveal three basic approaches that focus on achieving syntactic and semantic interoperability of user models: a shared-format approach, a conversion approach, and an intermediate approach.

The shared-format approach enforces the use of a shared syntax and semantics to represent user models. On the other hand, the conversion approach, as does not use a shared representation for user models, employs appropriate methods to transform the syntax and semantics of the user model used in one system into those of another system. Finally, an intermediate approach integrates the advantages of both approaches to enable adaptability in describing user models and to offer a mapping of user information from one system to another [7].

3.1 Shared-Format Approach

The use of a shared-format for the representation of user models has an obvious advantage. This is the acquisition by a DL system of user attributes discovered by other DL systems. In this way, the DL system may use the existing data for personalization without the user being obliged to input them again.

In the field of music information retrieval, Chai and Vercoe [9] proposed the representation of user models in a standardized format. Their User Model for Information Retrieval Language (UMIRL) uses the XML syntax in order to represent different user models that can be shared across systems. Another example from this field is the MPEG-21 standard [27] that defines an open framework for multimedia applications. Users are identified by their relationship to other users and every user may have specific rights and duties depending on his interaction with other users.

The user modeling community focuses on ontology-based approaches as the basis for the shared-format approach in order to achieve user model interoperability. Ontology-based approaches have several advantages that originate from the principles of this formalism. The ontological representation of user attributes allows the deduction of additional user attributes based on ontology relations, conditions, and restrictions. The use of an ontology-based user model increases the potential for user attributes to be shared among DL systems. Approaches that belong to this category and will be described in detail include the General User Model Ontology, the Unified User Context Model, and the Ontology based User Model.

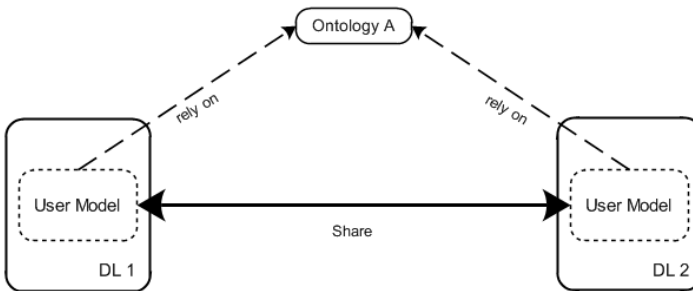


Fig. 1. Shared-Format Approach

General User Model Ontology. Heckmann et al. [13], [14], [17] proposed the General User Model Ontology (GUMO) in order to manage the syntactic and semantic variations between user modeling systems. The General User Model Ontology is based on OWL and is used for modeling user attributes and their interrelationships. The authors selected the user's attributes that are modeled in user-adaptive systems as well as user's interests and preferences. The construction of the user model ontology GUMO was based on the thought to divide the description of user model attributes into three elements: auxiliary - predicate - range. This description is called a *situational statement*. For example, the interests of a user in music could be described in the following way: auxiliary=*hasInterest*, predicate=*music*, and range=*low-medium-high*. The situational statement apart from the main information contains also contextual information and privacy preferences [12]. The privacy attributes (key, owner, access, purpose, retention) enable controlled propagation of sensitive information. Owner's intended privacy settings accompany the statement itself when it is exchanged among systems. User profile interoperability issues are handled by applying conflict resolution strategies among situational statements [14]. The authors also introduced the u2m.org user model service that is an application-independent server for maintaining and retrieving user profiles and for exchanging these profiles between different applications. A key advantage is that interoperability between distributed user-adaptive and context-aware systems is achieved because the semantics for all user model and context attributes are mapped to the general user model ontology GUMO [16].

The characteristics and attributes of GUMO are applied in the user model exchange language called User Modeling Markup Language, UserML [15], which promotes the exchange of user models across systems. The GUMO approach has been used for the representation of museum visitor's models in the Mobile Museums Guide [21] and it has been tested in a Positioning Service [5] and an Alarm Manager application [4].

The Unified User Context Model. Niederee et al. [25], [28] introduced a Unified User Context Model (UUCM) that can be used for modeling attributes of the user and his environment, i.e., the user context. Their proposal identifies two levels for the unified user context model, the abstract and the concrete level. The *abstract* level specifies the principal components of the UUCM that are: user context, user model attributes, main characteristics for attributes representation, and user model dimensions. For the cross-system personalization, this level specifies a shared ontology and all systems depend on this model. The *concrete* level defines a group of UUCM dimensions and attributes that include not only users' interests, but also tasks and relations to other entities and relevant user communities. Different attributes are modeled with the use of name/value pair. In this way, each attribute of the user context model is captured and new attributes can be easily added. Each UUCM attribute is represented by the following features: attribute name, attribute qualifier, attribute value, value qualifier, value probability, and attribute dimension. Moreover, the four UUCM dimensions selected for the context model are the Cognitive Pattern, the Task, the Relationship, and

the Environment. Finally, relevant subsets of the user context model are defined that are called user's working contexts and are used to distinguish the different roles that the user can play.

The UUCM defines the structure of the user context profile that can be used not only to express a user in a system but also as an intermediate type for the exchange of user profiles between different systems.

In the Niederee et al. approach for cross-system personalization, a context passport based on the UUCM is used to accompany the user when moves from a system to another. The context passport is a concise encapsulation of the user's current context profile and also includes the activities chosen by the user to be executed in order to complete the allocated tasks. When the user performs an activity, the respective part of the context passport is selected and used in order to better support the requirements of the user.

Ontology Based User Model. Razmerita et al. [31] proposed the Ontology based User Model (OntobUM) that is a generic ontology-based user modeling architecture developed for a Knowledge Management System (KMS). OntobUM was created within the Ontologging project [29] that aimed to implement the next generation of KMSs based on three technologies: ontologies, software agents, and user modeling.

OntobUM combines three different ontologies: the *user ontology* that expresses the users, the *domain ontology* that specifies the relationships between the different applications, and the *log ontology* that determines the semantics of user-application interaction. Semantic Web technologies were employed for the implementation of the user ontology, and the structure of this ontology was based on extended IMS LIP specifications.

The complete user model for a user is composed of an explicit part specified by the user via the user profile editor and by an implicit part retained by services. The explicit part of the user model encompasses attributes such as identity, email, address, abilities, cognitive style, and preferences. The implicit part is connected to experiences related to the use of the system. For this reason, the authors have enhanced the IMS LIP groupings by introducing the Behavior notion. The Behavior notion describes attributes of users interacting with a KMS such as level of activity, type of activity, and level of knowledge sharing. Based on users' activity in the system, OntobUM categorizes the users into three classes: readers, writers, or lurkers. These classes are properties of the type of activity attribute. The level of activity includes four characteristics that can be related to the user: very active, active, passive, or inactive. The level of knowledge sharing captures the level of acceptance of knowledge sharing methods. Based on the above attributes, the system is able to provide feedback and virtual rewards.

3.2 Conversion Approach

It is apparent that by adopting a shared format approach by DL systems, there are no syntactic or semantic heterogeneity issues to be solved. All the systems use the shared unified model that is easily exchangeable and interpretable. Nevertheless, the DL systems that exist nowadays are very heterogeneous and dynamic.

This makes it impractical, and in some cases even impossible, to use a shared user model and to enforce DL systems to adhere to a shared vocabulary. There is another approach, the conversion approach that does not compel DLs to use a unified model but defines appropriate mechanisms in order to transform the syntax and semantics of the user model attributes in one system into those used in another system.

An example of this approach is given in Stewart et al. [33], where the interoperability of user models between two different Adaptive Educational Hypermedia systems, MOT and WHURLE, is done via a one-to-one conversion. Firstly, the identification of a set of common attributes between the user models of the two systems is performed and then the conversion is completed through a peer-to-peer interaction. A more general approach is the Mediation of User Models approach proposed by Berkovsky et al. [3] that will be analyzed in the following section.

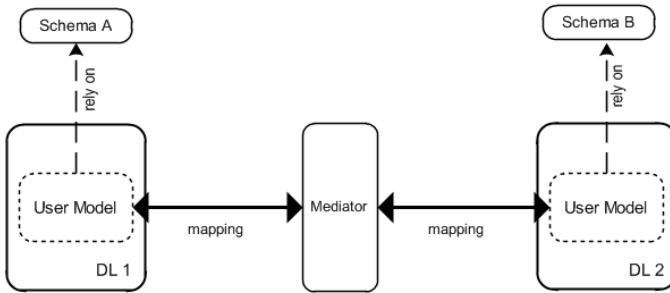


Fig. 2. Conversion Approach

Mediation of User Models. Berkovsky et al. [3] introduced in their work a generic framework for user model mediation. Mediation of user models is a process of transferring and incorporating the user model information collected by other systems for the goal of a particular recommendation proposition. This work also focuses on resolving the heterogeneity of the available user model information, as pays particular attention on the resolution of inconsistencies and conflicts among the information obtained from various systems.

A user model is represented in a three-dimensional space. The two generalized dimensions of this representation are *users* and *items*. These dimensions are called generalized because they may be described by sets of specific attributes. In order to facilitate provision of context-aware recommendation, the above two dimensions are extended by a third general dimension, indicating various contextual conditions and attributes that may be considered by the recommender system.

The mediation process includes a *target* recommender system that is the system requested to provide personalized recommendations to the user and one or more *remote* recommender systems that may provide pertinent user model data (past experiences) to the target recommender system.

The authors have defined four major types of mediation. The first type is called *cross-representation mediation* and is performed between experiences having the required values of all three attributes (user, item, context). The other three mediation types are mentioned as *cross-dimension mediations*. They are performed over the experiences having the required values of two attributes and a different value of one attribute. This means that the values of two out of three attributes are fixed and the mediation is conducted across the third attribute. Three types of cross-dimension mediations are distinguished: (a) *cross-user mediation*, where the values of item and context attributes are fixed and the user in the experiences is allowed to be revised; (b) *cross-item mediation*, where the values of user and context attributes are fixed and the item in the experiences is allowed to be revised; and (c) *cross-context mediation*, where the values of user and item attributes are fixed and the context in the experiences is allowed to be revised.

The authors resulted in the conclusion that mediation techniques may enhance the quality of the personalized recommendations and the performance of systems only in particular conditions. These conditions include the type of mediated data, the availability of user modeling information in the source and target systems, and many other factors. Therefore, the decision regarding applying the mediation should be taken only after a comprehensive analysis of these aspects.

The issue of user model mediation was studied within the SharedLife project [37] and the Passepartout project [1]. Furthermore, Berkovsky et al. [2] implemented user model mediation between a trip-planning system [32] and a personalized museum visitor's guide [23].

3.3 Intermediate Approach

The intermediate approach combines the benefits of the shared-format approach and the conversion approach. Specifically, the systems may use their own user models but they should provide a sharable part of their models in order to be exchanged with other systems. Then, a mediation method can be applied in order to provide a mapping of user information from one system to another. Concrete approaches that fall in this category are the Generic User model Component and the Framework for User Model Interoperability that will be presented in detail in the following sections.

Generic User Model Component. Van der Sluijs et al. [35], [36] introduced the Generic User model Component (GUC) that applies Semantic Web technologies to retain user models and to share user profiles between applications. Applications have the ability to store their user models in the GUC's application schema repository and to use GUC in order to upload user profiles that are valid only in a certain context. If the user uses the application in another context, another profile is stored. GUC employs the Shared User Model (S-UM), which includes the most used attributes within the domain, as a mean of user model exchange between various applications. S-UM can be used as a mediator for the exchange of user data between applications by creating a mapping to and from every application and S-UM.

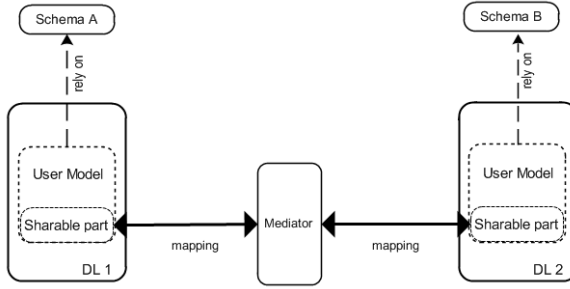


Fig. 3. Intermediate Approach

A schema mapping from a user model X to a user model Y includes a specification of how all attributes in user model X are mapped to corresponding attributes in Y . Schema mappings are produced by the GUC mapping component that needs the source and the target user model, say X and Y respectively. The mappings are created based on the similarities between two input user models and are expressed in the language SWRL. The mapping between user model X and user model Y has to be constructed only once and, therefore, can be created by a human designer. Irrespectively of the algorithm used for the schema mapping, the result must be examined and possibly be edited by hand before it can be used, because semantic structures may not be interchangeable on instance level.

The Generic User model Component deals also with the issue of data reconciliation when it is used for exchanging user data between applications. An application can ask for a specific user profile that is called a User Application View (UAV). A UAV of an application's user model can be translated into a (partial) UAV of another application's user model. Data reconciliation is supported by applying the OWL and SWRL techniques. For each application, rules should be defined to indicate how to reconcile data in the case of a conflict. Data reconciliation rules help to specify what to do if a value in the converted UAV already exists in the UAV that it should be incorporated. Possible approaches include the concatenation of the value with the current one, or the replacement of the current value, or the use of a given formula in order for a decision to be taken. Finally, GUC is able to apply privacy policies by allowing each user to have access to all data stored about him and control which applications may get access to which data.

Related research on Generic User model Component was conducted within the Alter-Ego project [34] and the IST MobiLife project [26].

A Framework for User Model Interoperability. According to Carmagnola's approach [7] systems do not require sharing a user model, but each system may use the user model that wish. Nevertheless, her approach uses RDF in order to assure the syntactic interoperability of the exchanged semantic-enriched user data. In her framework, the interoperability procedure takes place when an

application, called *receiver Rc*, may want to obtain data about a user from other systems, called *providers Pr's*. In order for a provider system to participate in the interoperability procedure, it should preserve a shareable user model that includes those parts of the user model that can be shared with other systems as RDF statements. Every statement represents the user model information that can be shared with other systems. A statement can be broken into four parts: a subject, a property, an object, and a value.

When a receiver system *Rc* needs to collect user model data from other systems, it begins the interoperability procedure first by retrieving the shareable user models of provider systems *Pr's* the user interacts with and then by searching for the particular user model data into those user models. Subsequently, receiver *Rc* obtains from a specific provider *Pr* the entire group of the statements that belong to user's shareable model. In order to measure the semantic similarity among the statements, each statement is divided into Object and Property. Then, the Object Similarity Algorithm is used to determine the similarity between the objects and the Property Similarity Algorithm is used to calculate the similarity between the properties. To compare the semantics of the objects in provider's and receiver's statements the author employs the Word Sense Disambiguation Theory which assumes that two terms are semantically interchangeable if their micro-contexts are interchangeable [18]. The micro-context of a term can be defined by dependence on two main sources of information: a) the information incorporated in the text or discourse in which the term appears, b) external knowledge sources, including lexical, encyclopaedic, etc.

Assuming the user is the same among two statements belonging to provider *Pr* and receiver *Rc* and the Osm Algorithm results that the objects in the specific statements are similar, it has also to be examined if the properties in those statements are similar. The Psm algorithm determines the similarity among the properties using the Levenshtein distance [24] that allocates a unit cost to all edit operations needed to transform one string into another.

Finally, the similarity measure between provider's *Pr* and receiver's *Rc* statements is obtained as the average of Osm and Psm results. The highest similarity measure between a provider's statement and the receiver's statement gives the highest relevance for the receiver *Rc*.

Carmagnola's research on user model interoperability was part of her doctoral thesis [6] that also contained a mechanism for the identification of the user whose information is shared across systems [8].

4 Discussion of Approaches

The approaches described in the previous section have been proposed as general user model interoperability solutions. They have not been specifically tailored to achieve user model interoperability in Digital Libraries. The six approaches presented in this work have the dynamics to be used for user interoperability in DLs; however they may have several weaknesses as they do not take into account the special needs and characteristics of the DL community.

Table 1. Comparison of Interoperability Approaches

	User Model Interop.	User Profile Interop.	User Context Interop.	Privacy Concerns
GUMO	+	+	+	+
UUCM	+	-	+	-
OntobUM	+	-	-	-
Mediation of User Models	+	+	+	-
GUC	+	+	+	+
Framework for UM Interop.	+	-	-	+

The above comparative table summarizes the six approaches, described in previous sections, in terms of user model, profile, and context interoperability and privacy concerns. The sign (+) indicates that the approach covers the specific requirement whereas the sign (-) indicates that the requirement is not captured. The General User Model Ontology (GUMO), the Unified User Context Model (UUCM), and the Ontology based User Model (OntobUM) that belong to the shared-format approach are suitable for systems that may easily agree to share a common user model format. In the case of several DLs interoperating, there may be special circumstances that do not allow a common model to be generally put into practice. Moreover, the GUMO approach pays attention to privacy by defining privacy attributes, whereas UUCM and OntobUM do not deal with this issue.

The approach that would probably be better suited for interoperable DLs should contain a form of conversion of one user model to another. An important approach is the Mediation of User Models. This approach, however, does not take into account the variety of user attributes the systems have collected about the user [22]. Also, this approach does not handle privacy issues. Another promising solution is the first intermediate approach, GUC, because not only it provides a schema mapping among different user models, but also, focuses on instance mapping among different user characteristics captured in different systems. The problem is that the mapping requires additional human effort and may not always be feasible.

The last approach that is presented in this work performs similarity checking among the various user model statements in order to overcome the semantic heterogeneity of different user models. Its weakness is that it does not provide a solution for the examination of the differences of the actual values captured in different systems.

5 Conclusion

In this paper we introduced the need for cross-Digital Library personalization by defining and analyzing user model, profile, and context interoperability. Then, we described six important user model interoperability approaches. Relevant

description was provided for those approaches that focus also on user profile and context interoperability as well as on privacy issues.

The conclusion that can be drawn from our investigation is that little work has been done on achieving user model interoperability across different systems. Moreover, there is a need for additional research on this issue in the Digital Library community in order to achieve cross-Digital Library personalization. New research directions are emerging that need not only to focus on user model interoperability, but also on reconciliation or consolidation of different user attributes as well as on propagation of access rights across different DL systems. Finally, more intensive efforts are needed to cope with the challenging issue of user context and its correlations with the user profile.

Acknowledgments. This work has been partially supported by the European Commission under FP7 Contract #231551 “DL.org: Digital Library Interoperability, Best Practices, and Modelling Foundations”.

References

1. Aroyo, L., Schut, H., Nack, F., Schiphorst, T., Kauw-A-Tjoe, M.: Personalized Ambient Media Experience: move.me Case Study. In: 12th International Conference on Intelligent User Interfaces, Honolulu, HI, pp. 298–301 (2007)
2. Berkovsky, S., Aroyo, L., Heckmann, D., Houben, G.J., Kröner, A., Kuflik, T., Ricci, F.: Providing Context-Aware Personalization through Cross-Context Reasoning of User Modeling Data. In: Workshop on Decentralized and Ubiquitous User Modeling, Corfu, Greece, pp. 2–7 (2007)
3. Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of User Models for Enhanced Personalization in Recommender Systems. *J. User Modeling and User Adapted Interaction* 18(3), 245–286 (2008)
4. Brandherm, B., Schmitz, M.: Presentation of a modular framework for interpretation of sensor data with dynamic Bayesian networks on mobile devices. In: LWA 2004, Lernen Wissensentdeckung Adaptivität, Berlin, Germany, pp. 9–10 (2004)
5. Brandherm, B., Schwartz, T.: Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 223–234. Springer, Heidelberg (2005)
6. Carmagnola, F.: From User Models to Interoperable User Models. PhD thesis, University of Turin, Italy (2007)
7. Carmagnola, F.: Handling Semantic Heterogeneity in Interoperable Distributed User Models. In: Kuflik, T., Berkovsky, S., Carmagnola, F., Heckmann, D., Krüger, A. (eds.) *Advances in Ubiquitous User Modelling*. LNCS, vol. 5830, pp. 20–36. Springer, Heidelberg (2009)
8. Carmagnola, F., Cena, F.: User identification for cross-system personalization. *J. Information Sciences* 179(1-2), 16–32 (2009)
9. Chai, W., Vercoe, B.: Using User Models in Music Information Retrieval Systems. In: International Symposium on Music Information Retrieval, Plymouth, MA, USA (2000)
10. Collet, M., Linton, F., Goodman, B., Farance, F.: Standard for learning technology-public and private information (PAPI) for learners (PAPI learner). IEEE P1484.2.1/D8, Draft (2001)

11. IMS Learner Information Packaging Information Model Specification, <http://www.imsglobal.org/profiles/lipinfo01.html>
12. Heckmann, D.: Distributed user modeling for situated interaction. In: GI Jahrestagung (1), Bonn, Germany, pp. 266–270 (2005)
13. Heckmann, D.: Ubiquitous User Modeling for Situated Interaction. In: Bauer, M., Gmytrasiewicz, P., Vassileva, J. (eds.) UM 2001. LNCS (LNAI), vol. 2109, pp. 280–282. Springer, Heidelberg (2001)
14. Heckmann, D.: Ubiquitous User Modeling. PhD thesis, Department of Computer Science Saarbrücken, University of Saarlandes (2005)
15. Heckmann, D., Kruger, A.: A user modeling markup language (UserML) for ubiquitous computing. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS (LNAI), vol. 2702, pp. 393–397. Springer, Heidelberg (2003)
16. Heckmann, D., Schwartz, T., Brandherm, B., Kroner, A.: Decentralized user modeling with UserML and GUMO. In: Workshop on Decentralized, Agent Based and Social Approaches to User Modelling, Edinburgh, Scotland, pp. 61–65 (2005)
17. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., Wilamowitz-Moellendorff, B.M.: GUMO – The General User Model Ontology. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 428–432. Springer, Heidelberg (2005)
18. Ide, N., Véronis, J.: Introduction to the special issue on word sense disambiguation: The state of the art. *Comput. Linguist.* 24(1), 2–40 (1998)
19. Kass, R., Finin, T.: A general user modelling facility. In: SIGCHI Conference on Human Factors in Computing Systems, Washington, USA, pp. 145–150 (1988)
20. Kobsa, A.: Generic user modeling systems. *User Modeling and User-Adapted Interaction* 11(1-2), 49–63 (2001)
21. Kruppa, M., Heckmann, D., Krüger, A.: Adaptive multimodal presentation of multimedia content in museum scenarios. *Künstliche Intelligenz* 19(1), 56–59 (2005)
22. Kuflik, T.: Semantically-enhanced user models mediation: Research agenda. In: 5th International Workshop on Ubiquitous User Modeling (UbiqUM 2008), Co-located with IUI 2008, Gran Canaria, Spain (2008)
23. Kuflik, T., Sheidin, J., Jbara, S., Goren-Bar, D., Soffer, P., Stock, O., Zancanaro, M.: Supporting Small Groups in the Museum by Context-Aware Communication Services. In: International Conference on Intelligent User Interfaces, Honolulu, HI, pp. 305–308 (2007)
24. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10, 707 (1966)
25. Mehta, B., Niederee, C., Stewart, A., Degemmis, M., Lops, P., Semeraro, G.: Ontologically-enriched unified user modeling for cross-system personalization. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 119–123. Springer, Heidelberg (2005)
26. MobiLife - life goes mobile, <http://www.ist-mobilife.org/>
27. MPEG-21, <http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm>
28. Niederee, C.J., Stewart, A., Mehta, B., Hemmje, M.: A Multi-Dimensional, Unified User Model for Cross-System Personalization. In: Workshop on Environments for Personalized Information Access, Gallipoli, Italy, pp. 34–54 (2004)
29. Ontologging CALT Project, <http://www.calt.insead.fr/Project/OntoLogging>
30. Orwant, J.: Heterogeneous learning in the Doppelgänger user modeling system. *User Modeling and User-Adapted Interaction* 4(2), 107–130 (1995)
31. Razmerita, L., Angehrn, A., Maedche, A.: Ontology-Based User Modeling for Knowledge Management Systems. In: Brusilovsky, P., Corbett, A., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 213–217. Springer, Heidelberg (2003)

32. Ricci, F., Arslan, B., Mirzadeh, N., Venturini, A.: ITR: A Case-Based Travel Advisory System. In: Craw, S., Preece, A. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 613–627. Springer, Heidelberg (2002)
33. Stewart, C., Cristea, A., Celik, I., Ashman, E.: Interoperability between AEH user models. In: International Workshop on Adaptivity, Personalization and the Semantic Web, Odense, Denmark, pp. 21–30 (2006)
34. Schuurmans, J., Zijlstra, E.: Towards a continuous personalization experience. In: 8th Conference on Dutch Directions in HCI, Amsterdam, The Netherlands, pp. 19–22 (2004)
35. van der Sluijs, K., Houben, G.: A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning* 16(1/2), 64–76 (2006)
36. van der Sluijs, K., Houben, G.J.: Towards a generic user model component. In: Workshop on Decentralized, Agent Based and Special Approaches to User Modelling, 10th International Conference on User Modeling, Edinburgh, Scotland, pp. 43–52 (2005)
37. Wahlster, W., Kröner, A., Heckmann, D.: SharedLife: Towards Selective Sharing of Augmented Personal Memories. In: Stock, O., Schaerf, M. (eds.) Reasoning, Action and Interaction in AI Theories and Systems. LNCS (LNAI), vol. 4155, pp. 327–342. Springer, Heidelberg (2006)
38. Wang, Y., Kobsa, A.: Privacy in Cross-System Personalization. In: Intelligent Information Privacy Management Symposium, Stanford, CA (2010)

An Ontology-Based Approach of Multimedia Information Personalized Search

Mihaela Brut and Florence Sedes

IRIT - Research Institute in Computer Science of Toulouse,
118 Route de Narbonne, 31062 Toulouse, France
{Mihaela.Brut,Florence.Sedes}@irit.fr
<http://www.irit.fr/>

Abstract. This paper discusses and provides a solution for the problem of adopting ontologies in order to model the users and the multimedia documents and to develop personalized search functionalities. First, the existing approaches that enable ontology-based semantic description of multimedia content are discussed. Then, current ontology-based solutions for personalized search functionalities inside adaptive hypermedia systems are presented. Our solution is exposed further, including the multimedia document model, the user profile development and the algorithmic solution that enables to provide personalized results to a user query.

Keywords: multimedia management, semantic annotations, ontologies, information indexing, personalized search.

1 Introduction

In the context of the multimedia information systems with an increasing number of available resources, the searching activity should be tailored to each user needs and interests. In order to be effective, the results provided to a user query should be provided according to the current user profile, which could include his/her preferences, tastes, backgrounds, knowledge or interests. In order to develop such personalized search facility, a matching between user queries, user profile and document representations should be accomplished.

This paper discusses and provides a solution for the problem of adopting ontologies in order to model the users and the multimedia documents and to develop personalized search functionalities. The main idea is to analyze the user queries according the ontology concepts and to execute them against the ontology-based documents metadata. In the beginning, the paper discusses the existing approaches that enable ontology-based semantic description of multimedia content are. Then, current ontology-based solutions for personalized search functionalities inside adaptive hypermedia systems are presented. Our solution is exposed further, including the multimedia document model, the user profile development and the algorithmic solution that enables to provide personalized results to a user query.

2 Ontology-Based Multimedia Content Description Approaches

In order to describe administrative, technical or physical features of the multimedia content, a lot of XML-based vocabularies were developed and standardized for different content types:

- *Images*: Exchangeable Image File Format (Exif)¹, IPTC Photo Metadata², VRA Core³, NISO Z39.87⁴, DIG 35⁵, PhotoRDF⁶.
- *Audio-visual content*: MPEG-7 (Multimedia Content Description Interface)⁷, MXF (Material Exchange Format)⁸, AAF (Advanced Authoring Format)⁹, ID3¹⁰, MusicBrainz¹¹, MusicXML¹², EBU P/Meta, MPEG 21;
- *Text*: TEI (Text Encoding Initiative)¹³.

The problem of semantically describing the content itself through a metadata-based layer of meaning such as to make the multimedia content semantics transparent to computer applications [1] could not be solved exclusively with the support of these vocabularies. Ontologies constitute the main instrument for developing such transparent semantic annotations of the multimedia content.

Ontologies are used mainly for two purposes with respect to the multimedia content:

- to provide semantic expression for the multimedia structural metadata expressed in XML vocabularies. As an integrant framework, the ontology provides in this case as well support for the interoperability issues between these vocabularies.
- to provide a semantic description of the multimedia content independently from the XML-based specialized vocabularies, e.g. based on domain ontologies.

While our solution is developed from the second purpose perspective, we present further some existing approaches following the both purposes: the domain ontology-based semantic descriptions have to be added up to the multimedia structural semantic metadata.

¹ Exif Version 2.2, Japan Electronics and Information Technology Industries Association: http://www.digicamssoft.com/exif22/exif22/html/exif22_1.htm

² <http://www.iptc.org/IPTC4XMP/>

³ <http://www.vraweb.org/projects/vracore4/>

⁴ <http://www.niso.org/>

⁵ <http://xml.coverpages.org/FU-Berlin-DIG35-v10-Sept00.pdf>

⁶ <http://www.w3.org/TR/2002/NOTE-photo-rdf-20020419>

⁷ <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

⁸ <http://www.smpte-mxf.org/>

⁹ <http://www.aafassociation.org/html/techinfo/>

¹⁰ http://www.id3.org/Developer_Information

¹¹ <http://musicbrainz.org/MM/>

¹² <http://www.recordare.com/xml.html>

¹³ <http://www.tei-c.org/>

2.1 Ontologies That Integrate XML-Based Vocabularies

Some ontologies were developed that aim to provide integrative support for describing all the multimedia features no matter of their initially XML-based expression. Such solutions include *COMM* ontology (<http://comm.semanticweb.org/>), ABC [2] or *aceMedia* ontology (<http://www.acemedia.org/>), all of them being developed on the top of MPEG7 vocabulary. The differences between them concern the coverage of the entire MPEG7 specification, as well as the maintenance of the initial MPEG7 structure.

The *aceMedia Ontology Framework* [3] define an integrated multimedia annotation framework based on a core ontology (DOLCE), two multimedia MPEG-7 based ontologies (VDO - Visual Descriptor Ontology - and MSO - Multimedia Structure Ontology), as well as domain ontologies such as PCS (Personal Content Management) and CCM (Commercial Content Management) Ontologies.

DELOS II Network of Excellence [4] defined an MPEG-7 upper ontology, which was extended with Semantic User Preference Description ontology and harmonized with MPEG 21 DIA Ontology, as well as with *SUMO* and *DOLCE* core ontologies in order to acquire an integrated annotation framework. *GraphOnto* was adopted as visual ontology-based annotation tool for multimedia content.

The goal of *COMM (Common Ontology for Multimedia)* ontology is to describe the semantics of multimedia content in terms of current semantic Web languages [5]. The *COMM* ontology exploits and extends the structure of the MPEG-7 specifications in order to provide support for organizing the multimedia metadata; *COMM* ontology provides also support for expressing all the multimedia features covered by the MPEG-7 specification, which forms a really huge set. The advantage of its formal semantics consists in enabling these features' expression, independently of the XML-based vocabulary through which the features were initially expressed. In other words, *COMM* provides support to express all the XML-based multimedia metadata having synonyms in MPEG-7 specification.

2.2 Ontology-Based Multimedia Content Semantic Description Approaches

Some specialized multimedia ontologies were also developed in order to capture and express the high-level semantics for multimedia objects: *aceMedia Visual Descriptor Ontology*, *mindswap Image Region Ontology*, *MSO - Multimedia Structure Ontology*, *VDO - Visual Descriptor Ontology*, *AIM@SHAPE* ontology for representing, modeling and processing knowledge which derives from digital shapes, *Music Information ontology*, *Semantic User Preference Ontology* developed to be used in conjunction the MPEG-7 MDS Ontology and with domain ontologies, in order to interoperate with MPEG-7 and allow domain knowledge utilization, *CIDOC CRM* core ontology for all multimedia objects, especially concerning cultural heritage items and events.

As could be noticed, there is a lot of support for expressing and organizing the multimedia semantic metadata, but not a generally accepted integrated framework. In order to manually develop such semantic metadata, different frameworks and tools were developed. Because of the high cost involved by this operation, multiple approaches to automatically obtain such semantic metadata were also developed, mainly around some concrete multimedia systems. We present below some representative approaches, where obtaining semantic metadata constitute an important step in integrating multimedia content in various personalized or customized functionalities. We present further some important existing examples.

In [1], a video content annotation architecture built on PhotoStuff image annotation tool¹⁴ is used to link MPEG-7 visual descriptors (obtained through automatic multimedia processing) to high-level, domain-specific concepts. The manually obtained multimedia semantic metadata is further used in order to improve the browsing and searching capabilities.

In [6] is presented a system where the multimedia content is annotated through three ontologies: the developed ontology on the top of MPEG-7, and two domain-specific ontologies. In order to enable the semantic interoperability, the three ontologies are merged with the support of ABC top-level ontology [2]. Alongside with the manual annotation, domain-specific inferencing rules are defined by domain-experts through an intuitive user-friendly interface in order to automatically produce supplementary semantic metadata.

In METIS project [7], the multimedia content is organized into a database, characterized by customizable media types, metadata attributes, and associations, which constitutes a highly expressive and flexible model for media description and classification. The multimedia ontology-based annotations could be also defined, due to the developed plug-in for the open-source Protégé ontology editor. The authors provide as case study the implementation of an archive system for research papers and talks in the Computer Science domain, classified according the ACM classification system. The semantic annotations are developed by the users, via a Web annotation interface. Scientific resources are thus available for browsing, classification, and annotation through the standard METIS Web administration interface.

The project aceMedia adopts manual ontology-based multimedia annotations, with the support of M-OntoMat-Annotizer. As well, the project developed a multimedia analysis system for automatically annotate the multimedia content based on the developed aceMedia Visual Descriptor Ontology. The system includes methods that automatically segment images, video sequences and key frames into a set of atom-regions while visual descriptors and spatial relations are extracted for each region [3]. A distance measure between these descriptors and the ones of the prototype instances included in the domain ontology is estimated using a neural network approach for distance weighting. Finally, a genetic algorithm decides the labeling of the atom regions with a set of hypotheses, where each hypothesis represents a concept from the above mentioned

¹⁴ <http://www.mindswap.org/2003/PhotoStuff/>

domain ontology. This approach is generic and applicable to any domain as long as specific domain ontologies are designed and made available.

As could be noticed, there are multiple ontology-based modeling solutions that enable to adopt an ontology-based description of the multimedia content. Mainly, some of them enable to express the common multimedia document features, such those mentioned in Section 2.1. In addition, some more detailed semantic descriptions of the multimedia content, expressed through domain ontology, are obtained manually or into a semi-automatic manner that exploits some inference rules or classification algorithms.

We will present further some representative approaches where the multimedia semantic annotations are considered for multimedia retrieval functionalities. Moreover, some approaches are outlined that consider user characteristics when responding to his queries.

3 Using Ontologies for Developing Multimedia Retrieval and Personalized Search Functionalities

3.1 Ontology-Based Multimedia Retrieval

The existing retrieval mechanisms implement an efficient ranking algorithm applied to the results provided for a certain query. Many ranking methods were introduced, based on clever term-based scoring, link analysis, evaluation of user traces etc. [8].

In [9] the MPEG-7 OWL ontology¹⁵ is used as upper-level multimedia ontology where three different music ontologies have been linked in order to annotate the multimedia content. System architecture is proposed that facilitates multimedia metadata integration and retrieval.

In SAFIRE project [10] MPEG-7 structure is used as basis for organizing multimedia features. Alongside with automatically extracted features, the semantic annotations are accomplished manually, using WordNet ontology in order to acquire disambiguate annotations. These annotations are exploited together with their synonyms for increasing the efficiency of the further query process.

In [11], ontology is used in order to define the video database model. Such ontology must be previously developed for a certain modeled domain, containing definitions of objects, events and concepts in terms of attributes and components. The system applies in a first phase a set of automatic multimedia processing techniques in order to segment the video into regions, and to extract features for each region (color, shape, color distribution etc.). If some regions have similar properties for a period of time (consecutive keyframes), the possible occurrence of an object could be inferred. By using similarity functions, objects identified from regions are assigned to their actual names by using information gained from the training set developed by experts according the considered ontology. The ontology-based data model enables the system to support ontology-based queries,

¹⁵ <http://rhizomik.net/ontologies/mpeg7ontos/>

able to specify objects, events, spatio-temporal clauses, trajectory clauses, as well as low-level features of objects.

In [12], a method is proposed for searching a document collection via queries that are constituted by ontology concepts. The ranking algorithm considers these concepts as distinct key-phrases, while the ontological relations are not exploited.

[13] describe a system with ontology-based annotation and retrieval capabilities for managing the audiovisual information. After the multimedia segmentation process, the annotations are made by specialists, by making reference to some previously selected ontologies, and stored in the semantic base. The search mechanism, implemented as an API, provides support for semantic queries, based on the some provided search templates.

As could be noticed, in the various frameworks are developed that exploit multimedia metadata mainly for a better information retrieval, that do not take into account the particularities of the user that accomplishes the search: for a specific query, the same results are provided to all users. Different approaches that consider user characteristics when responding to a user query were developed in the area of adaptive hypermedia systems, and some of them consider ontology-based content descriptions.

3.2 Ontology-Based Personalized Search

Given a particular user keyword-based query, the personalized search systems provide results that are tailored to the preferences, tastes, backgrounds and knowledge of the user who expressed it [14]. In systems that adopt ontological modeling, retrieving documents for a certain user query means in fact querying documents by the ontology concepts included into the query and filtering them based on the user model.

[15] describe Bibster, a Semantics-Based Bibliographic Peer-to-Peer System, which uses ACM ontology together with SWRC ontology in order to describe properties of the scientific publications. The retrieval mechanism makes also use of a learning ontology, developed on the fly, in order to reflect the actual content of the individual users. ACM ontology was also used, together with SWEBOK ontology in order to refine the e-learning materials annotation [16].

[17] develop an ontology for reformulating and storing the user queries in a semantic enriched form; in order to approximate the meaning of users' queries each query term is mapped to a Word-Net sense. The retrieval mechanism computes the similarity of documents and the already constructed query ontology, by using the AUTOMS5 proposed method that combines lexical, semantic, and structural matching methods.

[18] define a conceptual architecture for a personal semantic Web information retrieval system. The user requirements are reflected by his/her preferences, profile and constraints along with a query. A formal query is composed of three types of element fields: user preferences (UPs), content query (CQ) and Web service query (SQ). The responses combine Web content relevant to the query, but also information about the Web services potentially relevant to the user.

In general, the personalized search systems develop the user profile in terms of the history of user's keyword-based queries, correlating it with the document annotations. Personalization process could consider the user profile into one of three moments: during the retrieval process, in a distinct re-ranking activity or in a pre-processing of the user query [14].

We will expose further our personalized search solution were a same ontology is adopted in order to model the user query, the user profile and the document content.

3.3 Ontology-Based Solution for Personalized Search Inside Multimedia Systems

We present further an ontology-based solution for developing customized responses to the user queries. Domain ontology is adopted for modeling the user query, the user profile and the multimedia documents, as well as for locating pertinent result documents for the user query. The solution capitalizes some previous work while bringing also some new contributions.

Considering the medical domain and the MESH¹⁶ as example ontology, we will adopt a vector representation of its $n=25.588$ main concepts ("preferred terms"), alongside with an OWL ontology representation¹⁷ that enables to store as well the relations between concepts. Let us designate as $C[i]$, $i=1,n$, the vector that stores the MESH concepts. The n dimension will be further adopted for multiple vectors containing weights of MESH ontology concepts (the ith element correspond to the $ith@$ concept) and describe the document model, user query or user profile, as illustrated in the following sections.

Multimedia Documents Model and Representation. As we mentioned, in order to model the semantic metadata associated with multimedia documents, we need an ontology that provide support for expressing the common multimedia features, and one ore more domain ontologies through which the semantic of the multimedia content to be expressed. Harmonizing and integrating the both types of ontologies constitutes a problem itself.

Due to its extensive covering character of MPEG-7 descriptors, we consider the COMM ontology for representing the common multimedia features. As well, considering the particular case of medial domain, we adopt MESH ontology to describe the content itself of the multimedia content. We have to locate the better solution for binding these two ontologies.

COMM ontology uses DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)¹⁸ as a modeling basis, and it defines some modeling patterns [5]:

- *Decomposition*: exploits MPEG-7 descriptors for spatial, temporal, spatiotemporal and media source decompositions of multimedia content into segments.

¹⁶ <http://www.nlm.nih.gov/mesh/>

¹⁷ <http://bike.snu.ac.kr/sites/default/files/meshonto.owl>

¹⁸ <http://www.loa-cnr.it/DOLCE.html>

- *Annotation*: exploits the MPEG-7 very large collection of descriptors that can be used to annotate a segment. The annotations are associated to a particular media content region (or to the entire media document):
 - *Content Annotation*: for annotating the features of a multimedia document, which means for expressing its associated metadata (media-specific metadata/ For example, DominantColorAnnotation expresses the connection between a MPEG-7 DominantColorType with a segment.
 - *Media Annotation Pattern* - for describing the physical instances of multimedia content (general metadata). For example, MediaFormatType enable to express features such as FileSize=“462848”, FileFormat=“JPEG”;
 - *Semantic Annotation Pattern* (semantic metadata) - allow the connection of multimedia descriptions with domain descriptions provided by independent domain-specific ontologies.
- *Digital Data pattern* is used to formalize most of the complex MPEG-7 low-level descriptors.
- *Algorithm pattern* defines:
 - Methods - for the manual (or semiautomatic) annotations;
 - Algorithms - for automatically computed features (e.g. dominant colors) Every Algorithm defines at least one InputRole and one OutputRole which both have to be played by DigitalData.

As could be noticed, the Semantic Annotation Pattern acts as an interface between COMM and a domain-specific ontology (see Figure 1). It enables to include inside the COMM-based multimedia metadata some semantic metadata expressed through domain ontology concepts. We adopt this facility in order to integrate in the multimedia annotation the MESH concepts that describe the content of the current multimedia document. For representing these concepts, we adopt the technique exposed in the beginning of this chapter. For a specific multimedia document D_j , the MESH-based annotation are represented through a vector $D[j,i]$, $i=1,n$ ($n=25.588$), where each element $D[i,j]$ represents the weight of the concept $C[i]$ in the representation of the document D_j .

In [19] we presented a method for automatically obtaining this simplified representation in the case of textual documents. After a pre-processing phase, the terms frequency matrix associated to the document D_j suffers a dimension reduction through the latent semantic analysis technique: from a t -dimension corresponding to the detected t keywords, it is reduced to a k -dimension, where $k \ll t$. For each ontology concept, a t -dimensional vector representation is initially considered, which is reduced further to the same k -dimension. The distance between the dimensionally reduced concept vectors and the document vectors lead to detecting each concept weight for a document.

Images and audio-visual documents constitute a special challenge for indexing approaches because of their binary character [20]. However, some steps are done, and in we [21] exposed some preliminary results.

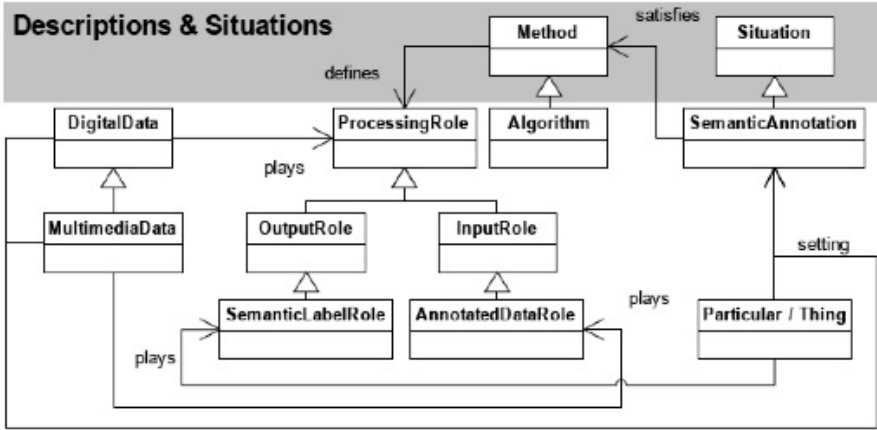


Fig. 1. Semantic Annotation Pattern in COMM ontology, according [5]

The explicit multimedia ontology-based annotation through a visual interface is the simplest but hence the most expensive method to acquire semantically enhanced metadata. Some specialized tools (such those further presented) were developed in order to support this type of manual annotation.

Protégé allows a user to load OWL ontologies, annotate data, and save annotation markup. *Protégé* provides only simple multimedia support through the Media Slot Widget, which allows general description of multimedia files like metadata entries, but not also description of multimedia document spatio-temporal fragments.

PhotoStuff allows annotating images and contents of specific regions in images according to several OWL ontologies of any domain (<http://www.mindswap.org/2003/PhotoStuff/>). Also designed for images, *Aktive Media* is an ontology based annotation system. *ImageSpace* provides support DAML+OIL language, and integrate image ontology creation, image annotation and display into a single framework.

ELAN (EUDICO Linguistic Annotator) provides support for linguistic annotation (analysis of language, sign language, and gesture) of multimedia recordings, including support for time segmentation and multiple annotation layers, but not the support of ontology. *OntoELAN* [22] extends *ELAN* with an ontology-based annotation approach: OWL linguistic ontologies could be used in annotations, while the ontological tiers should be linked to general multimedia ontology classes. With this role, *GOLD (General Ontology for Linguistic Description)* ontology is adopted [23].

3.4 User Query Processing

For a specified user query, the pertinent documents should be located for being provided as results.

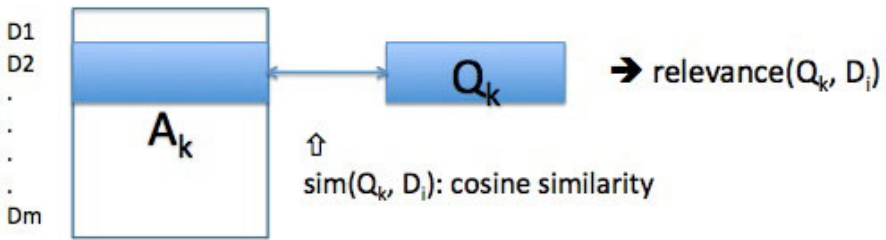
In order to facilitate the comparison between a user query and the available document models, we should represent the user query in a similar manner, namely a vector of concept weights. Thus, for a query Q , $Q[i]$ will mean the weight of the concept $C[i]$ in the query Q , where $i=1,25.588$. As example, a query like “pancytopenia in aids, workup and etiology” will be characterized by the MESH preferred terms “Pancytopenia - C15.378.700”, “AIDS-Related Complex - C20.673.480.080”, “Work - I03.946”.

However, such a concept-based vector representation is not obviously to be obtained. Remaining coherent with the document processing, we rely our technique on the query keywords. First, the relevant documents for the specified query are obtained:

- The query is represented first as a huge t -dimension vector $Q'[i]$ (where t is the same dimension as in the case of keyword-based document vectors);
- $Q'[i]$ receive value 1 for the positions corresponding to the query words, 0.5 for the positions corresponding to synonyms of the query-words (according Wordnet), and 0 on the others positions.

As example of synonymic variant for the specified we could mention “erythrocytes, leukocytes and platelets deficiency in Acquired Immuno-Deficiency Syndrome and pathogenesis”.

- The resulted $Q'[i]$ vector is reduced to the k dimension through the Singular Value Decomposition technique proper to the latent semantic analysis method.
- The pertinence of a document D_j for a user query Q will be given by the cosine similarity between their reduced k -dimension representations:



$$Q[i] = \frac{\sum_{j=1}^{10} D_j[i]}{10}, i = 1, 25588 \quad (1)$$

The first ten documents are selected having the biggest similarity to the current user query. However, in order to establish their order of pertinence for the current user, the user profile should be considered, as illustrated in the next section. Based on the determined top ten relevant documents for the specified

query, the concept-based query representation is obtained as the average of these documents' representations:

$$Sim(D_j, Q') = cos(D_j, Q') = \frac{D_j \cdot Q'}{|D_j| |Q'|} \quad (2)$$

3.5 User Profile Development

The adaptive hypermedia systems adopt a feature-based modeling technique, considering some important characteristics of the user as an individual: knowledge, interests, goals, background, and individual traits [24]. Three solution types were defined for modeling the user profile, based respectively on a keywords set, on a specific semantic network, or on a set of concepts belonging to multiple existing semantic networks, which could be taxonomies, topic maps, or even ontologies [24].

The user goals represent the most dynamic user characteristic since it illustrates his/her current activity, namely the run queries and the accessed documents among those returned as a query result. We adopt a user model that expresses the user current goals. The above presented vector representation for documents and queries enable to develop a similar user profile representation. It consists into a vector $U[i]$, $i=1,n$, where $U[i]$ represents the user interest degree concerning the concept $C[i]$, as deduced upon his provided queries and accessed documents.

- At the beginning of the working session, $U[i]=0$, $i=1,n$;
- When the user provides a query Q for expressing his current goals, the concepts mentioned by the query are included in his profile: $U[i] += Q[i]$, $i=1,n$;
- When the user accesses a document provided as a result to his query, the concepts that characterize the document D_j are also included in his profile: $U[i] += D[j,i]$, $i=1,n$;

It could be noticed that our solution consist in representing user goals as concept weights, while the user queries and accessed documents are not necessary to be stored. Their information is condensed into the $U[i]$ vector representing user profile.

3.6 Providing Customized Results to the User Query

We present further the steps of our algorithmic solution for for responding into a personalized manner to the user queries:

1. At the beginning of the working session, the user profile is empty: $U[i]=0$, $i=1,n$;
2. The user provides a query Q for expressing his current goals; the query is processed as it is described in Section 4.2;
3. The user profile is updated with the concept weights corresponding to the query: $U[i] += Q[i]$, $i=1,n$.

4. The similarity between this query and the representations of the available documents is calculated, and the list of documents with a similarity over a threshold τ is retained;
5. If this list is null, then the query Q is enriched by considering the parents of the component concepts, while reducing their weight with 50 percents (the degree of interest decreasing from a
6. If this list is not null, then it is re-ordered according the cosine similarity between user profile and each document vector. The list is displayed to the user;
7. When the user accesses a document provided as a result to his query, the concepts that characterize the document D_j are also included in his profile: $U[i] += D[j,i], i=1,n$;
8. When the user provides a new query, the elements of his current profile are divided by 2 in order to decrease the importance of his previous goals while emphasizing the goals expressed through the new query;
9. This query is considered by re-starting the step 2.

The ontology-based vector representations enable a very simple filtering process based on the cosine similarity between vectors, as Figure 2 illustrates.

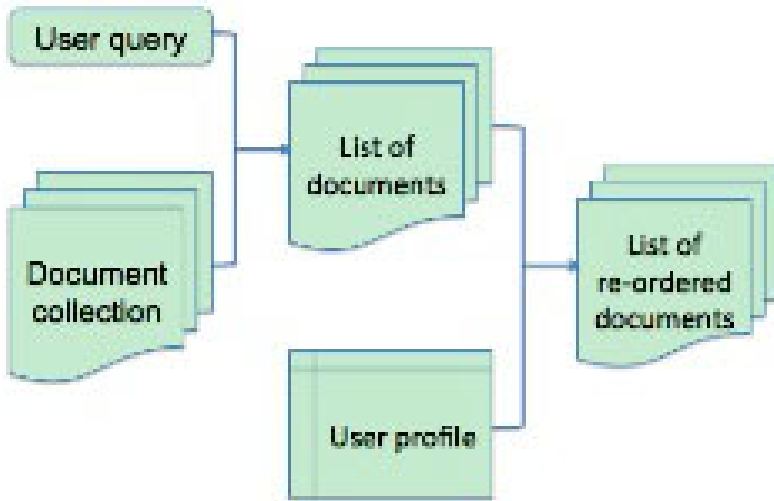


Fig. 2. The gradual filtering process based on ontological representation in the personalized search

3.7 Conclusions and Further Work

In this paper we presented a solution for the problem of adopting ontologies in order to model the user profile, the user queries and the multimedia documents. Despite some structured representations are available, are discussed and

are exploited for these resources, we adopted also a simplified vector representation that facilitates the matching and filtering processes that lead to the final personalized results list.

The present paper proposes a model that enables to semantically describe the multimedia content. This model is presented in the context of existing approaches that adopt ontologies in order to annotate multimedia resources. Its particularity consists in a simple solution for integrating domain ontology-based semantic annotations in the structure of the COMM ontology-based descriptions of the multimedia content, without requiring a special binding. Thus, the domain ontology considered in annotations is kept independently, while inside the COMM structure is included just information about the concepts and their weights for the current multimedia document.

The presented personalized search technique adopts the same domain ontology for representing user queries and for developing user profile.

We already worked in exploiting the semantic annotations associated with different resource types inside an existing tracking system that capture the user current activity, which is developed based on the Contextualized Attention Metadata (CAM)¹⁹ framework. In [26] we exposed a solution for recommending documents to users according to their current activity that is tracked in terms of semantic annotations associated to the accessed resources. We intend to extend this framework in order to handle the user query in the spirit of the presented semantic oriented approach. Tests using various multimedia collections are also considered for our future research explorations.

References

1. Song, D., Cho, M., Choi, C., Shin, J., Park, J., Kim, P.: Knowledge Representation for Video Assisted by Domain-Specific Ontology. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 144–155. Springer, Heidelberg (2006)
2. Lagoze, C., Hunter, J.: The ABC ontology and model (v3.0). *Journal of Digital Information* 2 (2001), <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Lagoze/>
3. Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., Hand-schuh, S., Kompatsiaris, I., Staab, S., Strintzis, M.G.: Semantic annotation of images and videos for multimedia analysis. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 592–607. Springer, Heidelberg (2005)
4. Tsinarakis, C., Polydoros, P., Christodoulakis, S.: Interoperability support for ontology-based video retrieval applications. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 582–591. Springer, Heidelberg (2004)
5. Arndt, R., Troncy, R., Staab, S., Hardman, L., Vacura, M.: COMM: Designing a well-founded multimedia ontology for the web. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 30–43. Springer, Heidelberg (2007)

¹⁹ http://www.ariadne-eu.org/index.php?option=com_content&task=view&id=39&Itemid=55

6. Hunter, J., Little, S.: A framework to enable the semantic inferencing and querying of multimedia content. *Int. J. Web Engineering and Technology* 2(2/3), 264–286 (2005)
7. King, R., Popitsch, N., Westermann, U.: METIS: a flexible foundation for the unified management of multimedia assets. *Multimed. Tools and Applications* 33, 325–349 (2007)
8. Long, X., Suel, T.: Three-level caching for efficient query processing in large Web search engines. In: *Proceeding of WWW 2005*. ACM Press, New York (2005)
9. Garcia, R., Celma, O.: Semantic integration and retrieval of multimedia metadata. In: *Proceedings of the Fifth International Workshop on Knowledge Markup and Semantic Annotation at the Fourth International Semantic Web Conference*, Galway, Ireland (2005)
10. Hentschel, C., Nurnberger, A., Schmitt, I., Stober, S.: SAFIRE: Towards Standardized Semantic Rich Image Annotation. In: Marchand-Maillet, S., Bruno, E., Nürnberger, A., Detyniecki, M. (eds.) *AMR 2006*. LNCS, vol. 4398, pp. 12–27. Springer, Heidelberg (2007)
11. Yildirim, Y., Yazici, A.: Ontology-Supported Video Modeling and Retrieval. In: Marchand-Maillet, S., Bruno, E., Nürnberger, A., Detyniecki, M. (eds.) *AMR 2006*. LNCS, vol. 4398, pp. 28–41. Springer, Heidelberg (2007)
12. Paralic, J., Kostial, I.: Ontology-based Information Retrieval. In: *Proc. of IIS 2003*, Croatia (2003)
13. Tsinaraki, C., Fatourou, E., Christodoulakis, S.: An Ontology-Driven Framework for the Management of Semantic Metadata Describing Audiovisual Information. In: Eder, J., Missikoff, M. (eds.) *CAISE 2003*. LNCS, vol. 2681, pp. 340–356. Springer, Heidelberg (2003)
14. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized Search on the World Wide Web. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 195–230. Springer, Heidelberg (2007)
15. Haase, P., Stojanovic, N., Volker, J., Sure, Y.: Personalized Information Retrieval in Bibster, a Semantics-Based Bibliographic Peer-to-Peer System. In: *Proc. of I-KNOW 2005*, Austria (2005)
16. Brase, J., Nejdl, W.: Ontologies and Metadata for eLearning. In: *Handbook on Ontologies*. Springer, Heidelberg (2003)
17. Kotis, K., Vouros, G.A.: Semantic Web Documents Retrieval through Ontology Mapping: Preliminary Results. In: *Proceedings of the 1st Asian Semantic Web Conference, ASWC 2006* (2006)
18. Yu, H., Mine, T., Amamiya, M.: An Architecture for Personal Semantic Web Information Retrieval System Integrating Web services and Web contents. In: *Proceedings of the IEEE International Conference on Web Services (ICWS 2005)*. IEEE Computer Society Press, Los Alamitos (2005)
19. Brut, M., Sedes, F., Dymitrescu, S.: A Semantic-Oriented Approach for Organizing and Developing Annotation for E-learning. *IEEE Transactions on Learning Technologies* 4 (2010)
20. Stamou, G., van Ossenbruggen, J., Pan, J.Z., Schreiber, G.: Multimedia Annotations on the Semantic Web. *IEEE Multimedia* (January-March 2006)
21. Brut, M., Sedes, F., Manzat, A.-M.: A Web Services Orchestration Solution for Semantic Multimedia Indexing and Retrieval. In: Barolli, L., Xhafa, F., Hsu, H.-H. (eds.) *Proc. CISIS 2009*, pp. 1187–1192. IEEE Computer Society, Fukuoka (2009)

22. Chebotko, A., Deng, Y., Lu, S., Fotouhi, F., Aristar, A.: An Ontology-Based Multimedia Annotator for the Semantic Web of Language Engineering. *International Journal on Semantic Web and Information Systems* 1(1), 50–67 (2005)
23. Farrar, S., Langendoen, D.T.: A linguistic ontology for the Semantic Web. *GLOT International* 7(3), 97–100 (2003)
24. Brusilovsky, P., Millan, E.: User models for adaptive hypermedia and adaptive educational systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)
25. Dolog, P., Schaefer, M.: Learner Modeling on the Semantic Web. In: *Proc. of PerSWeb 2005, Workshop on Personalization on the Semantic Web at 10th International User Modeling Conference* (2005)
26. Broisin, J., Brut, M., Butoianu, V., Sedes, F., Vidal, P.: A Personalized Recommendation Framework based on CAM and Document Annotations. In: *Proceedings of RecSysTel Workshop*, Elsevier, Procedia (2010)

Approaching Multimedia Retrieval from a Polyrepresentative Perspective

David Zellhöfer and Ingo Schmitt

Brandenburg Technical University Cottbus,
Department of Computer Science,
Database and Information Systems Group
`david.zellhoefer@tu-cottbus.de`

Abstract. Multimedia documents such as videos, images, or music are characterized by an amount of different qualities that can become relevant during a search task. These qualities are seldom reflected as a whole by retrieval models. Thus, we present a new query model, which fully supports the principle of polyrepresentation by taking advantage of quantum logic. We offer means to model document relevance as a cognitive overlap from various features describing a multimedia document internally. Using our query model, the combination of the aforementioned polyrepresentative features is supported by the mechanisms of a Boolean algebra. In addition, these overlaps can be personalized by user preferences during a machine-based learning supported relevance feedback process. The input for the relevance feedback is based on qualitative judgments between documents, which are known from daily life, to keep the cognitive load on users low.

We further discuss how our model contributes to the unification of different aspects of polyrepresentation into one sound theory.

1 Introduction

The creation of multimedia content such as videos, images, music, or rich-media content like Flash has been democratized. Content can be created and distributed without major burdens over the internet. Nevertheless, searching for appropriate multimedia documents is still complicated.

Multimedia documents are characterized by an amount of different qualities that can become relevant during a particular search task or context. In addition, the users' preference between these qualities may differ depending on the search goal. In one scenario, users may consider the hue of an image as very import, while they may prefer the artist during their next search. As a consequence, it is difficult to calculate the relevance of a multimedia document regarding a certain search task.

The aforementioned diversity of qualities of a multimedia document is reflected by its internal representation within a retrieval system. Multimedia documents are often split up into their atomic media types such as sounds or images.

These multimedia fragments can then be described with low-level features such as color or texture, high-level features such as textual annotations, or meta data like the creator or geo tag.

Common approaches for multimedia retrieval rely mostly on low-level features, because they can be extracted automatically in contrast to high-level features bearing more semantics. These content-based retrieval approaches are currently hitting a "glass ceiling", which cannot be overcome in order to improve their retrieval performance. For example, this effect has been shown in the music retrieval domain [1] but is present in other domains as well. To address this issue, user tags or annotations have been exploited. Though, annotations suffer from subjectivity, ambiguity, and noise. Additional media data such as Exif [2] can be exploited as well but it has a more relational characteristic, which makes it in particular feasible for the processing in database (DB) systems.

We rise to this challenge from the cognitive point of view. In the presented work, the relevance of a multimedia document with respect to a search will not be defined by an arbitrary combination of calculated similarity values based on low-level features or annotations. In contrast, we will model relevance as a *cognitive overlap* [2] of all different representations of a document neglecting if they have a traditional retrieval origin like histograms or text distances or a relational database origin such as a creation date or a media type. To motivate this different approach, we will outline the principle of polyrepresentation in information retrieval (IR). This section is followed by a discussion of a novel polyrepresentative query model based on results from quantum logic, which will be utilized to reflect multimedia documents and the search for them in a holistic way and to state user preferences amongst different aspects of these documents. We propose this query model as a reaction to the open question of how to form cognitive overlaps in a structured way [2]. Sec. 4 will present first experiments that will show the utility of the presented theoretical query model. At last, we will summarize our main contributions and pose open research questions.

2 Polyrepresentation in Information Retrieval

The principle of polyrepresentation has been introduced as a cognitive model in IR [3]. It is based on the hypothesis that different representations of a document can be utilized to form a cognitive overlap [2]. This overlap is an intersection of different functionally and cognitively different representations of a document, see Fig. 1. Here, *functionally different* representations are titles, abstracts, full texts, or the like. In contrast, *cognitively different* representations are depending on the user's current search task, personal preference or interpretations by a third person such as an indexer. Based on this overlap, the *probability of relevance* of a document w.r.t. a query, or more general: an information need, will be assessed. Because of the combination of a variety of functionally and cognitively different representations, it is assumed that the inherent uncertainty about relevance assessments in IR, i.e., the judgment an IR system makes about a document, will

¹ Exchangeable Image File Format; <http://www.exif.org>

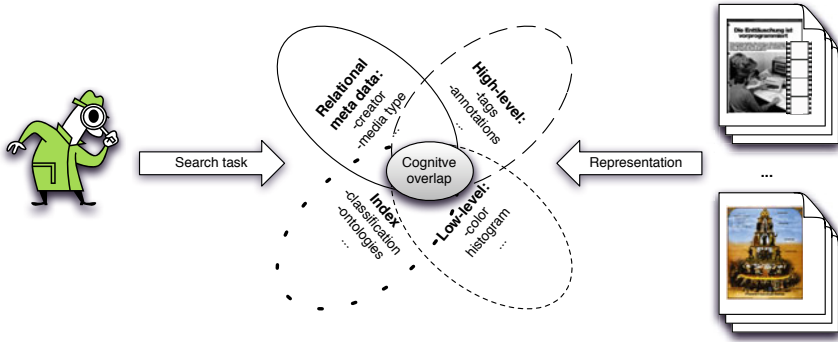


Fig. 1. Venn diagram of different document representations forming a cognitive overlap

be compensated – eventually improving the retrieval quality [3]. Skov et al. [4] strongly support this hypothesis.

Two other important findings from [4] are addressing the amount of representations that are utilized to form the cognitive overlap and the structure of the query that is used. It is stated that a high number of cognitively and functionally different representations indicating a relevant document improves the precision of the IR system. Another factor that is affecting precision is the amount of structure present in the query, which ultimately describes the cognitive overlap. Queries can range from *highly structured*, i.e., by supporting Boolean connectors that are used to formulate complex queries to *unstructured* such as bag-of-word queries. Skov et al. [4] show that highly structured queries perform better than queries with a decreasing degree of structure. These findings coincide with the results from [5,6]. In summary, a high number of features of a document combined with a highly structured query improves the retrieval performance.

Nevertheless, although the experimental results of the principle of polyrepresentation provide promising results, no frameworks have been suggested, which offer a means to model cognitive overlaps in a structured way. In the next section, we will introduce such a framework that also supports further personalization of cognitive overlaps.

3 A Polyrepresentative Query Model

Multimedia documents are intrinsically polyrepresentative due to the amount of different qualities they consist of and their internal representations, e.g., in functional space (low-level features such as histograms, high-level features such as annotations, or the aforementioned relational meta data) and in cognitive space, i.e., for a specific search task that is, e.g., affected by subjective preferences between these qualities. As said before, the similarity of multimedia documents to a given query depends strongly on the user’s notion of relevance.

This poses the question how all representations have to be combined, i.e., how a query should be stated in order to form a cognitive overlap expressing this notion of relevance. Current approaches in IR tackle this problem empirical for a particular domain, but the problem remains unsolved for arbitrary domains. Larsen et al. [2] see a need for further research on forming cognitive overlaps "depending on domains, media, genre, and presentation styles". Though, they do not address the influence of the user's subjective notion of relevance explicitly. As a consequence, the cognitive overlap for a search task is likely to vary from user to user although mainly contributing representations, which form the overlap may stay the same (see Sec. 3.2).

Recently, Frommholz and van Rijsbergen [7] discovered the issue of a missing framework for supporting the principle of polyrepresentation as well. Like our contribution, their work is based on results from quantum mechanics and logic [8]. However, they do not address the exploitation of structural queries to model the cognitive overlap in order to calculate the probability of relevance for a document.

To construct reasonable cognitive overlaps, we propose the usage of our query model. The model consists of a query language, which will be discussed in the following subsection, and an adaptive relevance feedback (RF) [9] process for user interaction and further personalization (see Sec. 3.2).

3.1 An Outline of the Commuting Quantum Query Language

The presented approach relies on the *commuting quantum query language* (CQQL) [10], which is based on results from quantum logic. It provides a unifying framework combing similarity-based retrieval conditions with relational conditions. Thus, it is capable of utilizing all possible access paradigms that are used to process and describe multimedia documents, in other words: IR and DB. Additionally, CQQL forms a Boolean algebra and is therefore a highly structured query language with all benefits that have been described in [4,5,6]. This property of CQQL is crucial as "structured Boolean-like query configurations will best support polyrepresentation in IR" [2]. Note that the usage of a Boolean algebra does not mean a re-introduction of the well-known drawbacks of the Boolean model into IR such as non-ordered result sets or a lack of expression for term/condition importance. CQQL uses similarity respectively probability values in order to calculate relevance such as the vector space model [11]. Examples of how CQQL can be used to construct cognitive overlaps are given throughout the following sections.

For the sake of completeness, we will outline the underlying concepts from quantum logic and mechanics and how they relate to the principle of polyrepresentation. In quantum mechanics, each microscopic system's state can be described by a normalized *state vector* $|\varphi\rangle$ [2]. Each state corresponds to a Hilbert

² We will use the Dirac notion for vectors, which is common in quantum mechanics. Here, $|x\rangle$ denotes a column vector and $\langle y|$ a row vector. $\langle x|y\rangle$ would express their scalar product.

space \mathbf{H} – a real-numbered vector space with a scalar product for the sake of simplicity. In our scenario, we interpret one state vector as one possible representation of a document, e.g., a low-level feature. To measure the state of a system, we rely on the simplified measurement given by *projectors*. A projector $p = \sum_i |i\rangle\langle i|$ is a symmetric and idempotent linear operator defined over a set of orthonormal vectors $|i\rangle$. Multiplying a projector with a state vector means to project the vector onto the respective vector subspace, i.e., calculating the probability that the state vector is relevant to the cognitive overlap. The probability of an outcome corresponding to a projector p and a given state vector $|\varphi\rangle$ is defined by:

$$\langle\varphi|p|\varphi\rangle = \langle\varphi|(\sum_i |i\rangle\langle i|)|\varphi\rangle = \sum_i \langle\varphi|i\rangle\langle i|\varphi\rangle$$

Hence, if we interpret this measurement as a probability value, the probability value equals the squared length of the state vector after it has been projected on the subspace spanned by the vectors $|i\rangle$. Furthermore, due to normalization the measured probability is equal the squared cosine of the minimal angle between probability value, furthermore, equals geometrically the squared cosine of the minimal angle between $|\varphi\rangle$ and the subspace represented by p . Tab. 1 summarizes the current findings.

Table 1. Relation of concepts between CQQL and polyrepresentation

CQQL	Polyrepresentation
state vector	representation
projector	cognitive overlap
quantum measurement	probability of relevance regarding the cognitive overlap

Because of the scope of this paper, we will refer to [10] for the necessary normalization steps and proofs and continue our discussion with the resulting algebraic evaluation of a CQQL query. We will use the query to model the cognitive overlap of all representations.

Let $f_\varphi(d)$ be the evaluation of a document d w.r.t. to a representation φ and $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, and $\neg\varphi$ logical connected representations (*conditions*), i.e., to what probability the document overlaps with one or a combination of representations present within the cognitive overlap. If φ is an atomic condition, $f_\varphi(d)$ forms the base case for the evaluation of a condition and results in a numeric value from the interval $[0, 1]$ for d , i.e., the probability of relevance for one representation of a document. For instance, this value can be derived from a normalized similarity measurement between the representation of the query and a document. Subsequently, the evaluation of the CQQL conditions is performed by recursively applying the succeeding formulas until the base case is reached:

$$\begin{aligned}
f_{\varphi_1 \wedge \varphi_2}(d) &= f_{\varphi_1}(d) * f_{\varphi_2}(d) \\
f_{\varphi_1 \vee \varphi_2}(d) &= f_{\varphi_1}(d) + f_{\varphi_2}(d) - f_{\varphi_1}(d) * f_{\varphi_2}(d) \text{ if } \varphi_1 \text{ and } \varphi_2 \text{ are not exclusive} \\
f_{\varphi_1 \vee \varphi_2}(d) &= f_{\varphi_1}(d) + f_{\varphi_2}(d) \text{ if } \varphi_1 \text{ and } \varphi_2 \text{ are exclusive} \\
f_{\neg\varphi}(d) &= 1 - f_{\varphi}(d).
\end{aligned}$$

A disjunction is called *exclusive* if it has the form $(\varphi \wedge \dots) \vee (\neg\varphi \wedge \dots)$ for some φ . It becomes clear that every evaluation is performed by simple arithmetic operations while being consistent with the laws of a Boolean algebra. Due to the disjunctive normal form (see [10]), every evaluation can be expressed as a sum of products of atomic condition evaluations on object attribute values. Hence, we can model a cognitive overlap using an arbitrary amount of representations in a structured manner. This overlap can then be evaluated using simple arithmetics. Another important finding is the relation of the evaluation rules of CQQL shown above to Kolmogorov's axioms of probability theory, which stresses the relation of probabilistic theory and quantum measurements:

$$\begin{aligned}
P(X \cap Y) &= P(X) * P(Y), \\
P(X \cup Y) &= P(X) + P(Y) \text{ (for mutually exclusive events), and} \\
P(X \cup Y) &= P(X) + P(Y) - P(X \cap Y)
\end{aligned}$$

Whereas P is the probability of an event X or Y . The negation is defined analogously.

Another characteristic of CQQL is its capability to include weights. A feature that will become important for the personalization of cognitive overlaps in Sec. 3.2. CQQL incorporates weights in order to express different importances of conditions within a query while staying consistent with a Boolean algebra [12]. Comparable approaches [13] can only apply weights outside their logic.

Weights in a weighted CQQL query q_θ are directly associated with logical connectors. Consider the following example of a weighted conjunction on different conditions φ_i : $q_\theta = (\varphi_1 \wedge_{\theta_1, \theta_2} \varphi_2) \wedge_{\theta_3, \theta_4} \varphi_3$. Weights are then replaced syntactically by constant values according to the following rules:

$$\begin{aligned}
\varphi_1 \wedge_{\theta_1, \theta_2} \varphi_2 &\rightsquigarrow (\varphi_1 \vee \neg\theta_1) \wedge (\varphi_2 \vee \neg\theta_2) \\
\varphi_1 \vee_{\theta_1, \theta_2} \varphi_2 &\rightsquigarrow (\varphi_1 \wedge \theta_1) \vee (\varphi_2 \wedge \theta_2)
\end{aligned}$$

Note that the usage of a logical query language does not mean the end of bag-of-words queries because such queries can be expressed by our approach as well by neutralizing all weights within a query by setting them 1. Eventually, an example query in the multimedia domain can be imagined as follows:

Example 1. Find all documents that are similar to a given image Img and that have been created in 2009.

$$\begin{aligned}
q_\theta := & (year = 2009) \wedge_{\theta_1, \theta_2} (edges \approx Img.edges \wedge_{\theta_2, \theta_3} \\
& colorLayout \approx Img.colorLayout \wedge_{\theta_4, \theta_5} \\
& (blue \approx Img.blue \vee_{\theta_6, \theta_7} orange \approx Img.orange))
\end{aligned}$$

The creation year is considered a Boolean condition, e.g., derived from meta data in this scenario, while all other conditions are retrieval conditions expressing similarity by distance measures on low-level features. All conditions are regarded as different aspects or representations of a document. The weights θ_i are utilized as a means of expressing importance of the different conditions.

3.2 Personalization and User Interaction

A problem, which is usually neglected while modeling the cognitive overlap between all feasible representations of a multimedia document, is the subjectivity of user preferences. While one user may be very sensitive to a certain texture that has to be present within an image, another might be interested in the presence of an associated annotation and a particular hue of the image. Although the general system of how the cognitive overlap is determined using a structured query based on various functionally and cognitively different representations might be similar, it is likely that different users do not agree on how important these contributing representations are. Consider the following example:

Example 2. A museum’s collection of multimedia documents shall be searched. The collection consists of digitized paintings, sculptures, photographs and books. The media type of a document is stored within a relational DB in addition to attributes such as creator, year of creation, historical location, or the like. Furthermore, low-level and high-level features are available to a retrieval system. Users are given the opportunity to search for documents about the Renaissance. In this scenario, the cognitive overlap will be formed by several functionally and cognitively different representations of a document, e.g., the century of creation, the media type, the creator, the cultural period, and low-level features depending on the media type as well as annotations or subject. A sample CQQL query condition could look like this:

$$\begin{aligned}
 q := & \neg(\text{mediaType} = \text{"photography"}) \wedge (\text{century} \approx 15) \wedge \\
 & ((\text{mediaType} = \text{"painting"} \rightarrow \text{dominantColor} \approx \text{RGB}(\dots) \wedge \dots) \vee \\
 & (\text{mediaType} = \text{"sculpture"} \rightarrow \dots) \vee (\text{mediaType} = \text{"book"} \rightarrow \dots)) \wedge \\
 & (\text{subject} \approx \text{"madonna"} \vee \text{subject} \approx \text{"pope"} \vee \dots) \wedge \dots \quad \boxed{3}
 \end{aligned}$$

Because of the structural power of CQQL, it is possible to formulate implications (\rightarrow) expressing a special set of conditions that has to be evaluated in case of a certain media type. Hence, it contributes to the solution of multi-domain problem stated in [2]. Here, low-level features such as dominant color or textures are used for a functional representation of images while other procedures are applied to different media types. Note that photographs are not queried as they were not existent during the Renaissance. According to this query, a similarity to the

³ For those readers who are familiar with CQQL: If a query condition such as $q := x \approx A \vee x \approx B$ is given, then $q \equiv \exists t : (t = A \vee t = B) \wedge x \approx t$ holds, i.e., the application of rules from first order logic to transform the query condition into a form that can be evaluated by CQQL.

query can be calculated for all documents in the collection, thus constructing their cognitive overlap.

Although the general construction of the cognitive overlap for a Renaissance search is intuitive, users are likely to have subjective preferences amongst many cognitive or functional representations. For example, in one scenario the century will be the most important, while in another the user will rely more on certain colors⁴ or subjects the document deals with. The most intuitive way to personalize the cognitive overlap is to use weights for all contributing representations. Weighting has a long tradition in IR since the early days of extended Boolean retrieval and has shown positive effects on user satisfaction and personalization [5,14,15]. Nevertheless, the weights have to be determined on an empirical basis or have to be set explicitly by the user, which can be a complicated task and needs knowledge of the underlying query model. As an example for arbitrarily chosen weights, [4] use weights to aggregate different features in order to form a cognitive overlap. In the presented approach, the weights are applied during the aggregation of different similarity values. This approach has the separation of the internal query logic from the weighting scheme in common with Fagin/Wimmers' approach [13], which is well-known in the DB domain. On the contrary, CQQL can include weights within its logic, i.e., all logical connectors can be associated with weights (θ_i), which will determine the impact of a condition on the query evaluation (see Sec. 3.1). This will change the example from above accordingly:

$$q_\theta := \neg(\text{mediaType} = \text{"photography"}) \wedge_{\theta_1, \theta_2} (\text{century} \approx 15) \wedge_{\theta_3, \theta_4} \\ ((\text{mediaType} = \text{"painting"} \rightarrow \text{dominantColor} \approx \text{RGB}(\dots))\dots$$

Here, the evaluation of the weights will happen during the algebraic evaluation of the query preceded by a logical transformation step (see Sec. 3.1), which is also described in detail in [16]. The integration of weights directly in the query language gives us the flexibility to adjust them according to the user perception, while maintaining the logical properties of the initial unweighted query. If the weights have been fixed before on an empirical basis or if they are part of the aggregation, an adaption is not easily possible.

Nevertheless, setting these weights can be a complex task for the user. The user needs knowledge of the query language and clearness about the subjective preferences between all conditions – both increasing the cognitive load for the user. In order to guarantee ease of usage, we support the user with a machine-based learning relevance feedback process. The core idea of the presented UI approach is to offer qualitative preferences as an input means to users. After an initial query on the document collection, possible relevant documents are shown to the user. If the user is not satisfied with the result, the query can be altered in an iterative manner. This is done via *inductive preferences* [17] amongst result documents, i.e., the user can express a preference between two objects such as object *A* is more relevant than *B*. See Fig. 2 for an illustration of the case that document #4 is better than image #3. Note that these judgments are known

⁴ Note that oil colors that were used in the Renaissance age in a particular way making it possible for an expert to estimate the cultural origin of a painting.



Fig. 2. Elicitation of an inductive preference between image 3 and 4

from daily life where one has to decide often between objects whose qualities are not known in detail. Hence, no additional cognitive burden is imposed on the user because of the familiarity with such judgments. Additionally, no further knowledge of the underlying query language is required. Hereby, we face the problem of eliciting expressive information about the user’s search goal without improving the cognitive load. Note that this cannot be said about all RF techniques [18].

After each user interaction iteration, the specified preferences, which will form a directed graph on the result documents serve as input for a machine-based learning algorithm that adjusts the weights of the given CQQL query according to the user needs. The learning algorithm is based on the downhill simplex algorithm [19], which can be used for solving non-linear optimization problems. Because of the algebraic evaluation rules of CQQL (see Sec. 3.1), we have to deal with this problem class in order to find weight values for a given CQQL query. A detailed discussion of the algorithms for a general CQQL weight learning scenario can be found in [16,20]. In case of errors, the algorithm communicates problems such as impossible preference combinations. A prominent example for such conflicts are preference cycles, e.g., if a user stated that document $A > B > C$ but $C > A$. These conflicts can be easily detected by a topological sorting of the aforementioned graph. Depending on the case, an automatic resolution via a prioritization or a Pareto composition of preferences or a manual reversal of the causing action is possible [21]. Another possible solution is the relaxation of the poset, e.g., by an automatic aging of the preferences comparable with the ostensive model approach [22]. Due to this learning step, it becomes possible to adapt the cognitive overlap on a user basis. As a consequence, a subjective cognitive overlap that is tailored to the user’s intentions is formed in an iterative manner by our approach. This overlap is based on a highly structured query language supporting weights. To improve the query outcome, the query can even be pre-formulated by domain experts or determined empirically. The weights within the query can then be considered as a means of steering the trend of the query results. Thus, they provide a valuable instrument for personalization.

4 Experimental Results

Although the main scope of the paper is theoretical, we conducted an experiment to gain some first insights into the utility of the discussed polyrepresentative

query model. In addition, we tested the approach in a real-world application. The experiment is based in the image retrieval domain using a document collection of holiday photographs taken in Indonesia. The cognitive overlap, i.e., the weighted CQQL query, is modeled as a weighted conjunction of various low-level features provided by the LIRE library [23]. In order to get a broad choice of representations of a document, the following features were chosen for the experiment: *SCALABLECOLOR*, *COLORLAYOUT*, *EDGEHISTOGRAM*, *AUTOCOLOR-CORRELOGRAM*, *COLORHISTOGRAM*, *GABOR*, *TAMURA*, *CEDD*, and *FCTH*. While the first features are known from the MPEG-7 descriptor set, the latter two are aggregated low-level features and are referenced in [23]. In the experiment, we did not rely on additional high-level features or the like because we wanted to measure the impact of preference elicitation on the presented query model. Fig. 4 (A) depicts the ranked results based on the aforementioned cognitive overlap with an initial neutral weighting, i.e., all representations contribute equally to the overlap⁵.

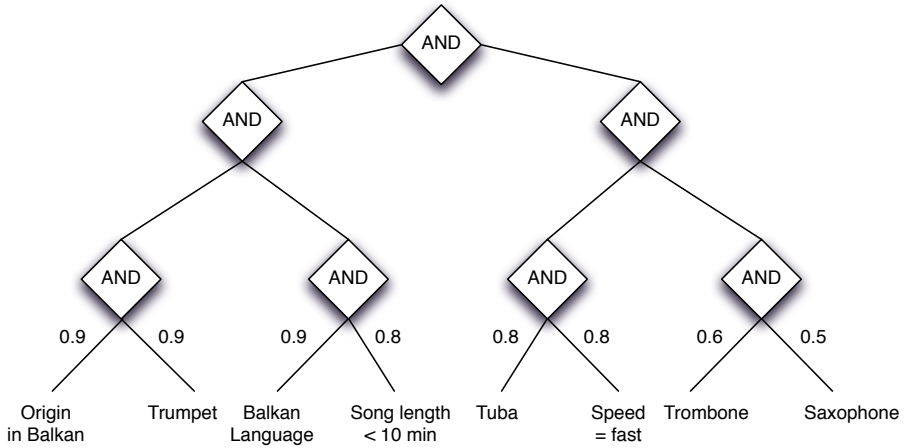


Fig. 3. Excerpt of the CQQL condition tree for gypsy brass music. The associated weights for the conjunctions are depicted along the edges.

For the first iteration, one preference was input. Here, image #4 (“pet dog”) was preferred less than image #5 (“temple”). After the interaction, the learning algorithm was executed with this constraint. Fig. 4 (B) shows the resulting rank. It becomes obvious that more temple images become visible in the rank excerpt. This finding is reflected by Fig. 5 (top). Here the rank error between the initial rank and the first iteration is depicted. The black diagonal lines shows the rank position after the iteration, while the red points show the initial rank position

⁵ Please note that it is not necessary to start with a neutral weighting. For instance, the possible inclusion of user profiles or other weight settings has been discussed in [24].



Fig. 4. Excerpt of the result list after 2 iteration steps. (Arrows illustrate applied preferences).

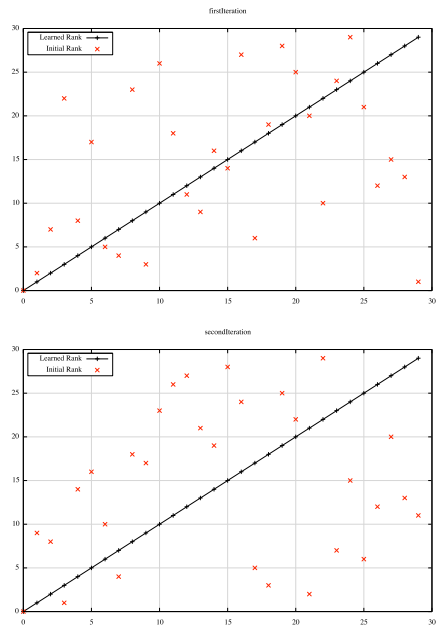


Fig. 5. Rank error for 1st (top) and 2nd iteration (bottom) based on preferences used in Fig. 4

of the same document. For instance, the document, which is ranked #5 (5 on the x-axis) after the preference modification has been ranked #17 before (see y-axis). Fig. 4 (C) illustrates the altered rank after the second iteration. In the second iteration two more preferences were added. See Fig. 5 (bottom) for the corresponding rank error.

To conclude with, the first conducted experiment shows that the presented polyrepresentative model is usable and can be examined in more depth. In combination with the iterative preference collection it is capable to incorporate the user's subjective notion of relevance and fit the cognitive overlap to it.

In addition to the lab experiment, we have tested the presented approach in a real-world scenario. As an example for music retrieval, CQQL has been used in the "GlobalMusic2one" project⁶ [25]. The goal of this project is to develop a music retrieval system for world music based on low-level data, user tags, folksonomies, and rule-based classifications of music genres, which were developed by domain experts from the Humboldt University of Berlin. During the test, cognitive representations such as timbre, genre etc. and different functional representations such as present musical instruments, artist, annotations, or low-level features for segments of a song were available to construct cognitive overlaps for various search tasks such as a similarity search for songs or to assistant during genre classification. The genre classification is based on weighted CQQL queries expressing domain expert knowledge that relies on relational data (e.g., song length) and low-level data such as present instruments. See Fig. 3 for a sample CQQL query being used for the classification of the genre "gypsy brass". For the sake of simplicity, all conditions are left in a textual representation. Internally, they are converted into numeric values.

Based on subjective assessments from domain experts, the ongoing field test shows that CQQL outperforms the plain low-level approach by far. We conclude that this is due to the structural power of CQQL that can reflect the polyrepresentative concepts being inherent in music.

In summary, the current results are very promising. Though more experiments are needed to provide resilient results, they already show that a logic-based combination of representations from different retrieval domains is possible and leads to an improved retrieval quality. The inclusion of weights supports the personalization of a search, thus improving user satisfaction.

5 Conclusions

This paper presents a query model for multimedia retrieval. The discussed approach addresses the current issues of retrieval quality by utilizing a cognitively motivated retrieval model. The suggested model consists of the query language CQQL, which is derived from quantum logic, and a user interaction method relying on relevance feedback and machine-based learning. In our approach, CQQL is used to model cognitive overlaps from different functional or cognitive representations of multimedia documents, i.e., applying the principle

⁶ http://www.globalmusic2one.net/en_summary.html

of polyrepresentation [3] for multimedia retrieval. It offers a new theoretical founded means to model cognitive overlaps in a structured way, which could also be extended to other retrieval domains. In this scenario, CQQL offers means to overcome limitations of other logical approaches such as fuzzy logic's dominance problem while guaranteeing the characteristics of a Boolean algebra [26].

The machine-based learning RF process is then used to personalize the cognitive overlap to improve user satisfaction. In order to keep the cognitive burden throughout the RF low, the user interaction is based on inductive preferences, which express "better-than" relations between documents. Further knowledge of the underlying, internal representations or the query model itself is not imposed on the user. Hence, the interaction is very intuitive. Our findings are supported by one lab experiment and a real-world scenario in which the suggested query model has been tested.

Fundamentally, some additional issues need further investigation. First, as we rely on the principle of polyrepresentation there is a strong relation of our approach to the polyrepresentation continuum [2]. Thus, a classification of our approach is due. Another challenge we are facing right now is the development of a complete graphical user interface to conduct resilient user tests, which are motivated by the promising first results in the aforementioned scenarios. We plan to reflect the principle of polyrepresentation in the user interface. This idea is supported by [18], who uses the principle for document visualization and navigation. We consider the visualization of multimedia documents as a special challenge because it has to reflect the underlying polyrepresentative nature of the documents. Thus, a plain display of a thumbnail or a video summary is not sufficient and needs further research.

In addition, we have to carry out further experiments, e.g., to measure precision and recall of our approach for different document collections. The main issue here is that our approach cannot be reflected very good by precision and recall alone because of the subjectivity that gets introduced into the retrieval by inductive preferences. These preferences allow a fine granular formulation of the user's subjective information need that goes beyond the traditional binary relevance scale, i.e., a clear relevance or irrelevance assessment of a document. We assume that the usage of the DCG measure [27], which relies on graded relevance assessments, will suit much better. Further, we plan to include usability and user experience measurements on a user basis as peers.

From a more theoretical point of view, we are extending the introduced RF process to formula learning. This means an application of a machine-based learning algorithm to find the needed cognitive overlap for a search task, i.e., learning a weighted CQQL query from user interaction alone. Here, we can rely on inductive preferences that are input by the user.

Another important finding that discriminates the quantum logic-based approach from the traditional vector space model is its capability to express so-called entanglements. Transferred to the IR domain, entanglements express the linkage of features or terms within a query, i.e., if a particular feature coincides with another and should therefore not be considered in isolation as it is usually

done within the vector space model. Potentially, this theoretic property can lead to semantically richer queries and a more realistic evaluation of them. Anyhow, this important aspect needs further research.

References

1. Aucouturier, J.J., Pachet, F.: Improving Timbre Similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences* 1(1) (2004)
2. Larsen, B., Ingwersen, P., Kekäläinen, J.: The polyrepresentation continuum in IR. In: *IiX: Proceedings of the 1st International Conference on Information Interaction in Context*, pp. 88–96. ACM, New York (2006)
3. Ingwersen, P., Järvelin, K.: *The Turn: Integration of Information Seeking and Retrieval in Context*. [Springer-11645 /Dig. Serial]. Springer, Dordrecht (2005)
4. Skov, M., Pedersen, H., Larsen, B., Ingwersen, P.: Testing the Principle of Polyrepresentation. In: Ingwersen, P., van Rijsbergen, C., Belkin, N. (eds.) *Proceedings of ACM SIGIR 2004 Workshop on "Information Retrieval in Context"*, pp. 47–49 (2004)
5. Hull, A.D.: Using Structured Queries for Disambiguation in Cross-Language Information Retrieval. In: *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval Electronic Working Notes*, pp. 24–26 (1997)
6. Turtle, H., Croft, B.W.: Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.* 9(3), 187–222 (1991)
7. Frommholz, I., van Rijsbergen, C.: Towards a Geometrical Model for Polyrepresentation of Information Objects. In: *Proc. of the "Information Retrieval 2009" Workshop at LWA 2009* (2009)
8. van Rijsbergen, C.: *The Geometry of Information Retrieval*. Cambridge University Press, Cambridge (2004)
9. Rocchio, J.: Relevance Feedback in Information Retrieval. *The SMART Retrieval System*, 313–323 (1971)
10. Schmitt, I.: QQL: A DB&IR Query Language. *The VLDB Journal* 17(1), 39–56 (2008)
11. Salton, G., Wong, A., Yang, S.C.: *A Vector Space Model for Automatic Indexing*, Ithaca, NY, USA (1974)
12. Schmitt, I.: Weighting in CQQL, Cottbus (2007)
13. Fagin, R., Wimmers, L.E.: A Formula for Incorporating Weights into Scoring Rules. *Special Issue of Theoretical Computer Science* (239), 309–338 (2000)
14. Salton, G., Fox, A.E., Wu, H.: Extended Boolean Information Retrieval. *Commun. ACM* 26(11), 1022–1036 (1983)
15. Lee, H.J., Kim, Y.W., Kim, H.M., Lee, J.Y.: On the Evaluation of Boolean Operators in the Extended Boolean Retrieval Framework. In: Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) *ACM/SIGIR 1993, Proceedings of 16th Annual International Conference on Research and Development in Information Retrieval*, Pittsburgh, USA, pp. 291–297 (1993)
16. Zellhöfer, D., Schmitt, I.: A Preference-based Approach for Interactive Weight Learning: Learning Weights within a Logic-Based Query Language. *Distributed and Parallel Databases* (2009)
17. Zellhöfer, D.: Inductive User Preference Manipulation for Multimedia Retrieval. In: Böszörmenyi, L., Burdescu, D., Davies, P., Newell, D. (eds.) *Proc. of the Second International Conference on Advances in Multimedia* (2010)

18. White, W.R.: Using searcher simulations to redesign a polyrepresentative implicit feedback interface. *Inf. Process. Manage.* 42(5), 1185–1202 (2006)
19. Nelder, A.J., Mead, R.: A Simplex Method for Function Minimization. *Computer Journal* 7, 308–313 (1965)
20. Schmitt, I., Zellhöfer, D.: Lernen nutzerspezifischer Gewichte innerhalb einer logik-basierten Anfragesprache. In: Freytag, C.J., Ruf, T., Lehner, W., Vossen, G. (eds.) *Datenbanksysteme in Business, Technologie und Web (BTW 2009)*, 13. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme (DBIS), Proceedings, Münster, Germany, March 2-6. *lni. GI*, vol. 44, pp. 137–156 (2009)
21. Zellhöfer, D.: Eliciting Inductive User Preferences for Multimedia Information Retrieval. In: Balke, W.T., Lofi, C. (eds.) *Proceedings of the 22nd Workshop "Grundlagen von Datenbanken 2010"*, vol. 581 (2010)
22. Campbell, I.: Interactive Evaluation of the Ostensive Model: Using a New Test Collection of Images with Multiple Relevance Assessments. *Inf. Retr.* 2(1), 89–114 (2000)
23. Lux, M., Chatzichristofis, A.S.: Lire: Lucene Image Retrieval: An Extensible Java CBIR Library. In: *MM 2008: Proceeding of the 16th ACM International Conference on Multimedia*, pp. 1085–1088. ACM, New York (2008)
24. Zellhöfer, D., Schmitt, I.: A Poset Based Approach for Condition Weighting. In: *6th International Workshop on Adaptive Multimedia Retrieval* (2008)
25. Schiela, K.: Ein CQQL-basiertes Musikretrievalsystem f. GlobalMusic2one BTU Cottbus: Master's Thesis. PhD thesis, Brandenburg University of Technology, Cottbus (2010)
26. Schmitt, I., Zellhöfer, D., Nürnberger, A.: Towards quantum logic based multimedia retrieval. In: IEEE (ed.) *Proceedings of the Fuzzy Information Processing Society (NAFIPS)*, pp. 1–6. IEEE, Los Alamitos (2008)
27. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20(4), 422–446 (2002)

Knowledge Based Multimodal Result Fusion for Distributed and Heterogeneous Multimedia Environments: Concept and Ideas

Florian Stegmaier¹, Tobias Bürger², Mario Döllner¹, and Harald Kosch¹

¹ Chair of Distributed Information Systems, University of Passau, Germany

² Knowledge and Media Technologies Group, Salzburg Research, Austria

Abstract. Distributed multimedia retrieval (DMR) is a key issue in today's information systems. One problem in DMR is the fusion of results retrieved from multiple locations, which is required in order to present the results in an integrated, consolidated and aligned form. This paper sketches a distributed multimedia retrieval scenario in which requirements, a conceptualization and an implementation of a knowledge-based data fusion approach is proposed. The approach is to be used together with the MPEG Query Format and is supported by benchmarks and classifications to derive knowledge used in result fusion.

Keywords: Multimedia retrieval, result fusion, benchmarking.

1 Introduction

Activities in storing and retrieving images within databases can be traced back to the late 1970s, where first conference contributions¹ introduced the use of relational databases for images⁴. In general, these early works focused on the annotation and retrieval of images by textual information. For this purpose, the images were described by keywords or prose descriptions and common relational database technologies and their text-based retrieval approaches were used for searching within the pool of annotated images.

Since then, the complexity and diversity in the domain of multimedia retrieval experienced a tremendous boost. First of all complexity increased by focusing not only on image data but also video and audio data emerged. Next, driven by Web 2.0 concepts and the common availability of camera attached smart phones, the amount of user generated and user annotated multimedia data rise continuously. This resulted to the well-known sites like Flickr², YouTube³, Picasa⁴, etc. This, and especially the digitalization within the TV sector demand for satisfying retrieval approaches over domains.

¹ E.g., Data Base Techniques for Pictorial Applications, 1979.

² <http://www.flickr.com/>

³ <http://www.youtube.com/>

⁴ <http://picasa.google.com/>

One key issue in this context is the fusion of results in a distributed multimedia retrieval scenario. Related to this, the contribution of this paper tackles for example the following questions: *How can external knowledge support the result fusion process? What kind of requirements and what kind of knowledge are beneficial and should be considered?* Based on the discussed ideas and concepts, a first implementation proposal is introduced covering a multimodal multimedia retrieval scenario in a distributed and heterogeneous environment.

The remainder of this paper is structured as follows: Section 2 discusses individual aspects and its complexity in multimodal result fusion. This is followed by Section 3, defining requirements for a knowledge based result fusion strategy. The generation of knowledge is introduced in Section 4. The injection of the extracted information into the result fusion process and a first implementation proposal are highlighted in Section 5 and Section 6, respectively. Related work in the area of result aggregation is introduced in Section 7. Finally, the article concludes in Section 8.

2 Result Fusion: Issue Characterization

A distributed multimedia retrieval system (DMRS) provides access to physically and maybe geographically detached data sources. In order to present a unified view of these different results sets, they have to be combined. This approach has been termed *result fusion* or *metasearch* in the Web Information Retrieval (IR) community, denoting techniques that combine pre-ranked results from multiple search engines into one consistent result. Montague and Aslam differentiate between two different types of metasearch techniques: *internal* and *external metasearch* [17]. External metasearch treats existing search engines which are potentially operating on diverse document sets as black boxes and consolidates their output. Internal metasearch engines combine multiple sub-engines that are operating over the same set of documents. This paper focuses on external metasearch engines, potentially combining multiple evidences based on a variety of modalities, which (can) operate over different sets of documents.

Typically, many modalities can be used to retrieve relevant data based on the type of query a user submits. Therefore, also the fusion of results from these modalities can be done on various ways. In an open environment, in which various types of queries can be issued, the fusion of results is difficult as evidence has to be available on how to combine the single results and which source of evidence on the correctness of a retrieved result item to trust. In this paper, the trust layer will be based on knowledge. By enabling the system to collect informations about the distributed environment, the current situation of fusing single results may be improved.

Practically, in a DMRS, each modality or distinct search engine used for retrieval acts as an expert generating separate result lists and similarity or rank scores. In order to get the best results for a query, the separate lists have to be combined into a single list by associating weights to each list. The weighted lists are then combined through the application of a particular fusion technique.

Traditional fusion techniques in IR can be divided into rank- and score based fusion methods. Score-based methods make use of similarity values computed for each retrieved document, while rank-based methods make use of the position of the document in the retrieved result list. One problem in combining results from distributed search engines is the learning of optimal combination of weights assigned to the independent engines. Techniques proposed in the literature either assign weights in advance or learn weights on the fly based on relevance feedback or data mining algorithms. Weights essentially express which search engine can handle which type of query better.

Important phases in result fusion include, depending on the applied strategy, the normalization of scores, the elimination of duplicates, the re-ranking of results, and the aggregation of the result lists.

3 Requirements for a Knowledge Based Result Fusion Strategy

As already mentioned in the previous sections, the quality of a multimedia retrieval process in the tackled domain heavily depends on the fusion of the single result sets. The requirements for the proposed result fusion strategy are defined as follows:

(i) Use of the MPEG Query Format (MPQF)

At present, the recently issued MPQF is the most specific query language for multimedia retrieval, as shown in [10]. Here, complex queries are built by the use of a rich set of multimedia specific query types (e.g., Query by Example), comparison types (e.g., greater than) and Boolean operators (e.g., AND). Therefore, a result fusion strategy for multimedia retrieval should at least support a subset of MPQF. The key advantage in implementing MPQF is the standardized, unified abstraction layer especially capable for distributed and heterogeneous environments.

(ii) Support of multimodality

Multimodality is an ambiguous concept. In this paper, multimodality in the terms of multimedia retrieval specifies a mixture of different types of multimedia data. Since MPQF is not closed to a particular multimedia data type (e.g., video) or metadata format (e.g., MPEG-7 [16]) the result set may be highly heterogeneous. The result fusion strategy should be able to process the present data types (e.g., for duplicate elimination) or unify diverse metadata formats (e.g., for presentation). Here, query classification techniques could help to overcome these issues, e.g., by analyzing the query condition tree or the desired output.

(iii) Estimating the quality of connected retrieval services

In distributed environments, it is most likely that retrieval services vary on the one hand in their offered retrieval functionalities (e.g., Query by Example vs. metadata based query) and on the other hand in their implementation details. The actual implementation may use different features to compare multimedia data or even base the similarity calculation on different metrics. Informations

about e.g., processing speed, precision or recall, could improve the retrieval process enormous. Such parameters could define an parametrizable evaluation function for retrieval service estimation.

(iv) Independence of underlying test data set

In addition to informations that can be collected or adjusted during query execution, it is an important task to have an initial knowledge about retrieval services before executing the first queries. For this reason, a multimedia specific benchmark consisting of a ground truth and the associated queries should be defined. Such an evaluation gives a first indication of the system retrieval behavior. Since most systems are soon tuned to a specific benchmark, a opportunity must be found to align different benchmarks to be independent.

(v) Knowledge-based algorithm selection

The benefit of (ii), (iii) and (iv) serve as the result fusion knowledge base and are leading directly to the fifth and most important requirement. By the use of the collected informations about the single retrieval services and query classification, the result fusion strategy should be enabled to choose the best set of algorithms of a given algorithm pool to process the single result sets.

(vi) Rank or score-based fusion

In order to fuse results in a distributed multimedia retrieval system, either similarity scores which have to be normalized, have to be available or rank values, which reflect the position of documents in the distinct result lists.

4 Knowledge Generation

In this paper, two types of knowledge sources will be considered: *external* and *internal sources*. As an external source for knowledge generation, Section 4.1 examines Content Based Multimedia Retrieval (CBMR) benchmarks introducing well-known substitutes and general concepts of benchmarking. Internal knowledge generation will focus on the query language itself, here MPQF. To this end, Section 4.2 investigates two different ways of classifying a query.

4.1 Benchmarking Content Based Multimedia Retrieval

In order to compare the effectiveness, performance and quality measurements of different algorithms and approaches, a benchmarking system is commonly used. For example, a well-known organization that has established benchmarks for relational databases is the Transaction Processing Performance Council⁵ (TPC). In addition, the multimedia retrieval community started encouraged projects: In the video domain, the Text REtrieval Conference⁶ (TREC) series initiated the TRECVID⁷ [22] workshop in 2003. Its main goal is to support information retrieval researcher by defining large sets of test beds including for example queries,

⁵ <http://www.tpc.org/>

⁶ <http://trec.nist.gov/>

⁷ <http://trecvid.nist.gov/>

a ground truth and scoring functions. Looking into the still image domain, ImageCLEF [18] is a famous substitute in creating benchmarks. This track started also in 2003 and belongs to the Cross Language Evaluation Forum⁸ (CLEF) aiming on the evaluation of image retrieval applications. For this purpose, ImageCLEF offers two main image collections (photographs and medical images) and corresponding query requests consisting of example images and text. Recently, the international standardization body JPEG⁹ also initiated a discussion¹⁰ about the need for a standardized still image benchmark as part of the recently issued JPSearch [12] project that defines abstract interfaces for an image retrieval framework. Though, this standard defines its own image specific query language, the JPEG Query Format (JPQF), which is a subset of MPQF. A call for requirements [7] has been published to establish a standardized benchmark environment. These ongoing research efforts combined with the multimodality present in multimedia data itself may indicate, that the creation of an unified multimedia benchmark is an open (maybe not solvable) issue. Therefore, the outcome of current research in this topic is the generation of domain specific benchmarks, tied to specific use cases.

In spite of the aforementioned differences of available benchmarks, they share the same basic four components [15]: a large amount of *test data (i)* is the basis for each meaningful evaluation. Mostly, it consists of a broad variation of the data in the represented domain (e.g., different genres of videos for a video test set). Correspondingly, *queries (ii)* specify the requests, whereas the tested systems respond with a certain subset of the test data. Finally, *qualitative evaluation metrics (iii)* calculate the distance (similarity) to the given *ground truth (iv)* for every request describing the retrieval quality.

Table I shows a variety of basic evaluation metrics/criteria that are commonly used to describe the retrieval ability of a system. These can be divided in technical as well as result set related criteria. Technical criteria mostly indicate the hardware setup of a system, like its connection speed or the scalability. In contrast to that, result set related criteria indicate, whether the items are correct or useful. In this field, precision [2] and recall [2] are the most prominent substitutes. A major drawback here is the possibility to manipulate them. For example one always gets a high precision value when retrieving a small number of items. Some solutions for this issue are proposed in the literature, like calculating the quality after a certain amount of items, see [19]. There also exist a category of measures, that also take the position of the result item and or similarity into account, e.g. normalized discounted cumulative gain (nDGC) [6] or relative weighted displacement (RWD) [20]. These measures might be very important in order to construct a global result set, which is the basic task for a external metasearch engine. In any case, it depends on the underlying benchmark, which measures can be calculated (heavily depending on the ground truth).

⁸ <http://www.clef-campaign.org/>

⁹ <http://www.jpeg.org/>

¹⁰ <http://www.jpeg.org/newsrel28.html>

Table 1. Set of evaluation metrics/criteria commonly used by benchmarks

<i>Evaluation criteria</i>	<i>Description</i>
Precision	proportion of retrieved material that is actually relevant
Recall	proportion of relevant material actually retrieved in answer to a search request
Binary preference measure [3]	measure of how often non-relevant items are misplaced
nDGC	measures the usefulness of a document based on its position in the result list
RWD	takes the similarity and the rank of an item into account (ideal weighted displacement is 0)
Scalability	systems ability to handle different portions of data related to the processing speed
Response time	needed time to answer a request
Connection Speed	uplink of the retrieval service to the network
Supported query types	indicates, whether the system is specialized or shares a broad scope of functionalities

Our idea is to calculate a specific benchmarking score for a system on the basis of the produced/derivable quality measures. This helps to compare different retrieval systems, potentially evaluated by different benchmarks. Here, we assume, that the benchmarks share a certain degree of overlap in the extracted measures. As already said, the most important information for an external meta-search engine (for reranking) is, whether a relevant item is well ranked in the result set or not.

In a first stage, the following formula can be used to calculate a benchmarking score:

$$bs_{es_i} = \sum_{i \in C} w_i \cdot v_i,$$

where w_i is a weighting factor, v_i a evaluation criterion, C the set of evaluation criteria and es_i the evaluated system.

The weighting factors of each evaluation criteria may be configured while creating the query (e.g., using an user interface) as well as by the use of machine learning approaches combined with query classification. Since relevance feedback is also integrated into MPQF, it could be also used to reflect the users opinion about the retrieved results into the calculated benchmark scores.

4.2 Classification of MPEG Query Format Requests

In our proposal, we distinguish between two ways of classifying a query. The goal of the first way is to generate complexity classes of MPQF queries. Here, the query structure, especially the query condition part, will be analyzed in order to determine the actual complexity of processing. Here, query complexity could be defined on the amount of Boolean operators in a query or the way, how the

different operands are assembled in the overall query. It would be possible to extract knowledge, which could serve as an input to adjust the weighting factors of the benchmarking affecting e.g. the re-ranking process. In order to start the investigation on this topic, the theoretical MPQF algebra is needed (yet under definition).

Our second way of classifying a MPQF query is focusing on the composition of the anticipated result. Inside a MPQF query, the *Output Query Format (OQF)* specifies a message container for query responses covering paging functionality and the definition of individual result items. The `ResultItem` element of the OQF holds a single record of a query result. Besides, the OQF provides a means for communicating global comments (by the `GlobalComment` element) and status or error information (by the `SystemMessage` element). This element provides three different levels for standardized signaling problems, namely `Status`, `Warning`, and `Exception`. These status codes now allow a preprocessing of the single result sets. If an error or a warning is present, the affected result set may be excluded from the result fusion or re-ranked by a penalty value. By analyzing the desired output of a query, more specifically the `ResultItems`, appropriate algorithms (e.g., for duplicate elimination) can be chosen. Finally, the `originId` attribute indicates the source service of the result. This attribute allows the adjustment to the results of the CBMR benchmarking.

In addition to the knowledge extraction, the aggregation process must ensure the semantic validity of the aggregated result set. Here, non semantic validity would imply, that informations encapsulated in the single results may be blurred or lost while fusing the single result sets. This issue has been discussed in [8] by evaluating OQF attributes with respect to a result aggregation.

5 Injection of Knowledge into the Result Fusion Process

This section combines the introduced result fusion stages (c.f., Section 2) with the benefits of the extracted knowledge (c.f., Section 4). Figure 1 illustrates an abstract result fusion process that has been injected by internal and external knowledge. Accordingly to requirement (i) (c.f., Section 3), this process takes several MPQF responses as an input and returns an consolidated MPQF response. Since the result sets may contain any types of multimedia data¹¹ - req. (ii) - appropriate algorithms must be selected to execute the phases of the result fusion (e.g., query classification for duplicate elimination). On the basis of the informations gathered by the benchmarking in the query classification particular algorithms will be chosen of an algorithm pool, see requirement (v). The system may now be adjusted to every incoming query that leads to a very flexible system. An useful example for this would be an emergency, where it is more important to retrieve the result fast than with the highest precision. This feature will be enabled by the parametrizable evaluation function, see req. (iii). In

¹¹ E.g., video, still images, metadata descriptions, ...

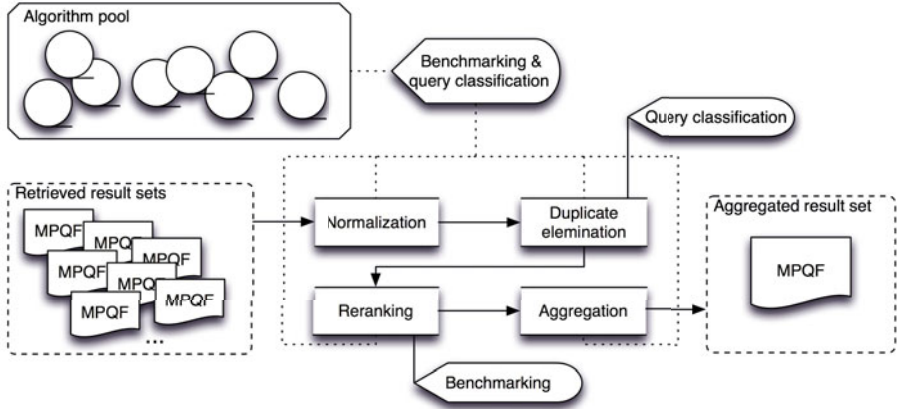


Fig. 1. Injection of knowledge into the result fusion process

particular the calculated benchmark score will highly affect the re-ranking. For example, the result items of the backend with the highest (unified) benchmark score - req. (iii) and (iv) - will be pushed up in the overall ranking / scoring, see requirement (vi).

6 Implementation Proposal

In a first stage, the implementation proposal will focus on the image retrieval domain using JPQF¹², more specifically it will be integrated into an interoperable image retrieval system. This system consists of the following components:

QUASI:A (Fig. 2a) is a JavaFX¹³ based query generation and result presentation tool. The main requirement for this user interface (UI) is usability. Any user should be able to create complex queries without expert knowledge. This is achieved by creating the query condition in a tree based manner. It features amongst others Query by Example, query by metadata as well as relevance feedback. These can be combined by Boolean operators, such as AND.

Observer (Fig. 2b) is a query visualization and tracing tool also implemented with JavaFX. It distinguishes between several processing units of a query inside AIR and presents useful informations of a particular phase (e.g., processing time).

AIR is a multimedia middleware framework [23], following the external meta-search paradigm (c.f., Section 2). This system is designed to operate in highly distributed and heterogeneous environments and implements subsets

¹² Using JPQF in the implementation proposal is no contradiction to the requirements, since it is a subset of MPQF, reduced to the image domain.

¹³ <http://www.javafx.com/>



(a) QUASI:A - Query generation



(b) Observer - Query tracing

Fig. 2. JavaFX based user interfaces

of MPQF as well as JPQF. The key advantages of the current implementation of AIR is the (a-)synchronous distribution / segmentation of queries by the use of a service discovery functionality and metadata transformation on the basis of JPSearch transformation rules [9].

A heterogeneous image retrieval environment contains two distributed image datastores offering retrieval functionalities. Here, heterogeneity is expressed by diverse query languages (e.g., SQL/MM or XQuery) and metadata formats (MPEG-7 and Dublin Core [11]).

In this system, AIR is acting as a mediator between the user interfaces and the storage layer, as the abstract architecture in Figure 3 indicates. At the moment, AIR is able to perform very limited result fusion tasks, such as aggregation of result sets only containing still images. It is not possible to aggregate multimodal sets, like a combination of still images and metadata informations, in a feasible way. In more detail, the Backend Benchmarking Layer is a only placeholder and the Response Layer is equipped with a Round-Robin approach [8]. These components should be implemented using the introduced concepts and ideas of the knowledge injected result fusion strategy.

Figure 4 shows a first attempt to design the integration of the benchmarking environment. Following the JPSearch standard, the (prospective) standardized benchmark will be integrated into the benchmarking layer. The workflow will be as follows: after a retrieval service registered at AIR, it will receive the ground truth and the query set by the use of a benchmark interchange format. Since the system is not closed to a particular benchmark, it is also possible that the retrieval service uses its own benchmark. After the particular benchmark has been performed, the set of evaluation criteria will be returned to AIR. Now the evaluation function will calculate the overall backend benchmarking score, which will serve as an initial measurement of the retrieval quality.

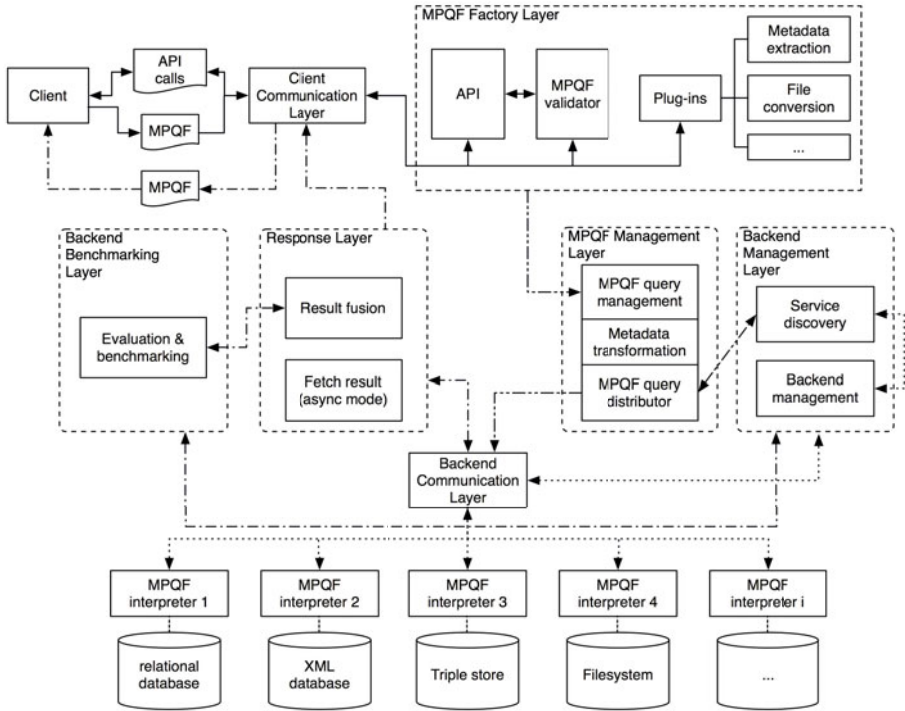


Fig. 3. Architectural overview of AIR

7 Related Work

There is a lot of research focusing on the combination of multiple queries issued on the same or different document collections. Related work in this area can be roughly grouped in two groups: (1) work whose aim is to integrate results on the same query from different search engines and (2) work whose goal is to improve retrieval performance by issuing different types of queries on the same document collection, potentially operating on different document representations. Early work on the combination of retrieval results includes experiments from Fisher and Elchesen, who showed that retrieval results were improved by combining two Boolean searches over two different document representations [13]. In 1997, Lee presented his hypotheses on conditions for successful result fusion which initiated a number of contributions in the research community [14]. Following Montague and Aslam, traditional fusion techniques can be divided into rank- and score based fusion methods [17] (cf. Figure 5).

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

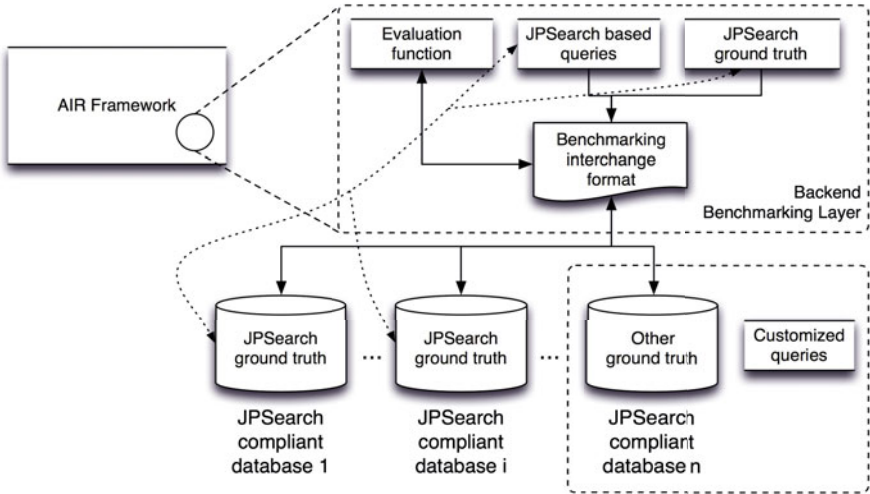


Fig. 4. Integration of benchmarking into the AIR multimedia middleware

or taking the minimum, maximum, or average of the retrieval scores [21, 14]. Rank-based methods include rCombMNZ which uses ranks instead of relevance scores or ReciprocalRankFusion which sums up the reciprocal of the rank of each result [5]. Voting-based methods were proposed by Montague and Aslam, first based on the Borda count, a positional voting algorithm, and later based on the Condorcet-fuse model, a majoritarian voting algorithm [1, 17].

Furthermore, specialized score-based fusion strategies were proposed for multimedia retrieval which include the linear combination [24] or the min/max aggregation of scores [25]. Others applied machine learning techniques to determine

	no training data	training data
ranks only	Condorcet-fuse	Weighted Condorcet-fuse
	Borda-fuse	Weighted rCombMNZ
	rCombMNZ	
relevance scores	CombMNZ	Weighted CombMNZ

Fig. 5. Classification of Result Fusion Techniques [17]

the weights for various retrieval strategies [26]. An evaluation of the adaptability of current result aggregation techniques regarding MPQF can be found in [8].

8 Conclusion and Future Work

This paper introduced concepts and ideas to enrich the result fusion process by additional knowledge, extracted from internal and external sources. The main contribution is the combination of benchmarking techniques and query classification with the result fusion problem in heterogeneous and multimodal multimedia environments. In addition, requirements have been formalized that should be fulfilled by implementations.

Currently, the work on the external knowledge extraction is already initialized with the main focus on the definition of the backend interchange format. After finalization, the implementation of the backend benchmarking layer will be started, along with the improvement of the evaluation function. In parallel, the elaboration about the query classification will be started.

Acknowledgement. This work has been partially supported by the THESEUS Program, which is funded by the German Federal Ministry of Economics and Technology.

References

1. Aslam, J., Montague, M.: Models for metasearch. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 276–284 (2001)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, 1st edn. Addison Wesley, Reading (1999)
3. Buckley, C., Voorhees, E.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 25–32 (2004)
4. Chang, N.S., Fu, K.S.: Query by pictorial example. *IEEE Trans. on Software Engineering* 6(6), 519–524 (1980)
5. Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 758–759 (2009)
6. Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, 1st edn. Addison Wesley, Reading (2009)
7. Döller, M.: Draft of Call for JPSearch Benchmark Requirements, ISO/IEC JTC 1/SC 29/WG1 N5399, JPEG (2010)
8. Döller, M., Bauer, K., Kosch, H., Gruhne, M.: Standardized Multimedia Retrieval based on Web Service technologies and the MPEG Query Format. *Journal of Digital Information* 6(4), 315–331 (2008)
9. Döller, M., Stegmaier, F., Kosch, H., Tous, R., Delgado, J.: Standardized Interoperable Image Retrieval. In: ACM Symposium on Applied Computing (SAC), Track on Advances in Spatial and Image-based Information Systems (ASIIS), Sierre, Switzerland, pp. 881–887 (2010)

10. Döller, M., Tous, R., Gruhne, M., Yoon, K., Sano, M., Burnett, I.S.: The MPEG Query Format: On the way to unify the access to Multimedia Retrieval Systems. *IEEE Multimedia* 15(4), 82–95 (2008)
11. Dublin Core Metadata Initiative. Dublin Core Metadata Element Set - Version 1.1: Reference description (2008), <http://dublincore.org/documents/dces/>
12. Dufaux, F., Ansorge, M., Ebrahimi, T.: Overview of JPSearch: a Standard for Image Search and Retrieval. In: 5th International Workshop on Content-based Multimedia Indexing (CBMI 2007), Bordeaux, France (2007)
13. Fisher, L., Elchesen, D.: Effectiveness of combining title words and index terms in machine retrieval searches. *Nature* (238), 109–110 (1972)
14. Lee, J.H.: Analyses of multiple evidence combination. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 267–276 (1997)
15. Marchand-Maillet, S., Worring, M.: Benchmarking Image and Video Retrieval: an Overview. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, Santa Barbara, California, USA, pp. 297–300 (2006)
16. Martinez, J.M., Koenen, R., Pereira, F.: MPEG-7. *IEEE Multimedia* 9(2), 78–87 (2002)
17. Montague, M., Aslam, J.: Condorcet fusion for improved retrieval. In: Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 538–548 (2002)
18. Müller, H., Geissbuhler, A.: Benchmarking image retrieval applications. In: Proceedings of the 7th International Conference on Visual Information Systems (2004)
19. Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recogn. Lett.* 22(5), 593–601 (2001)
20. Narasimhalu, A.D., Kankanhalli, M.S., Wu, J.: Benchmarking multimedia databases. *Multimedia Tools Applications* 4(3), 333–356 (1997)
21. Shaw, J., Fox, E.: Combination of multiple searches. In: Proceedings of the 2nd Text Retrieval Conference (TREC), pp. 243–252 (1994)
22. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and TRECVID. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 321–330 (2006)
23. Stegmaier, F., Döller, M., Kosch, H., Hutter, A., Riegel, T.: AIR: Architecture for Interoperable Retrieval on Distributed and Heterogeneous Multimedia Repositories. In: Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2010), Desenzano del Garda, Italy (2010)
24. Yan, R., Hauptmann, A.: The combination limit in multimedia retrieval. In: Proceedings of the 11th ACM International Conference on Multimedia, pp. 339–342 (2003)
25. Yan, R., Hauptmann, A., Jin, R.: Multimedia search with pseudo-relevance feedback. In: Proceedings of the International Conference on Image and Video Retrieval, pp. 238–247 (2003)
26. Yan, R., Yang, J., Hauptmann, A.G.: Learning query-class dependent weights in automatic video retrieval. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 548–555 (2004)

A Contour-Color-Action Approach to Automatic Classification of Several Common Video Genres

Bogdan E. Ionescu^{1,2}, Christoph Rasche¹,
Constantin Vertan¹, and Patrick Lambert²

¹ LAPI - University Politehnica of Bucharest, 061071 Bucharest, Romania

² LISTIC - Polytech'Savoie, B.P. 806, 74016 Annecy, France
{bionescu,rasche,cvertan}@alpha.imag.pub.ro,
patrick.lambert@univ-savoie.fr

Abstract. We address the issue of automatic video genre retrieval. We propose three categories of content descriptors, extracted at temporal, color and structural level. At temporal level, video content is described with visual rhythm, action content and amount of gradual transitions. Colors are globally described with statistics of color distribution, elementary hues, color properties and relationship. Finally, structural information is extracted at image level and histograms are built to describe contour segments and their relations. The proposed parameters are used to classify 7 common video genres, namely: animated movies/cartoons, commercials, documentaries, movies, music clips, news and sports. Experimental tests using several classification techniques and more than 91 hours of video footage prove the potential of these parameters to the indexing task: despite the similarity in semantic content of several genres, we achieve detection ratios ranging between 80 – 100%.

Keywords: video genre classification, action content, color properties, contour structural information, video indexing.

1 Introduction

An interesting challenge in the fields of content-based video indexing is the automatic cataloging of video footage into predefined semantic categories. This can be performed either globally, by classifying video into one of several main genres, e.g. cartoons, music, news, sports or even further into some sub-genres, e.g. identifying specific types of sports (football, hockey, etc.), movies (drama, thriller, etc.); or either locally by focusing on classifying segments of video such as retrieving concepts, e.g. outdoor vs. indoor, violence, action, etc. [1].

Being related to the issue of data mining, video genre classification involves two steps: *feature extraction* and *data classification*. Feature extraction and selection is one critical step towards the success of the classification task. The main challenge is to derive attributes discriminant enough to emphasize particularities of each genre while preserving a relatively reduced number of features. Most of the existing feature extraction approaches rely on visual elements, like color,

temporal structure, objects, motion, etc., which are to be used either alone or in combination with text or audio features. A complete state-of-the-art of the literature on this matter is presented in [2]. In the following we shall highlight several approaches we consider representative and related to our work.

For instance, [3] addresses the genre classification task using only video dynamics. Motion information is extracted at two levels: background camera motion and foreground or object motion and a single feature vector is constituted in the DCT transformed space. This is to assure low-pass filtering, orthogonality and a reduced feature dimension. A Gaussian Mixture Model (GMM) based classifier is then used to identify 3 common genres: sports, cartoons and news. Despite the simplicity of this single modal approach, it is able to achieve detection errors below 6%.

A more complex approach which uses spatio-temporal information is proposed in [4]. At temporal level, video content is described in terms of average shot length, cut percentage, average color difference and camera motion (4 cases are detected: still, pan, zoom, and other movements). Spatial features include face frames ratio, average brightness and color entropy. The genre classification task is addressed at different levels, according to a hierarchical ontology of video genres. Several classification schemes (decision trees and several SVM approaches) are used to classify video footage into movie, commercial, news, music and sports; movies into action, comedy, horror and cartoon, and finally sports into baseball, football, volleyball, tennis, basketball and soccer. The highest precision for video footage categorization is around 88.6%, for sports categorization it is 97% while for movie categorization it is around 81.3%, however no information is provided on the recall ratios.

An interesting true multi-modal approach, which combines several types of content descriptors, is proposed in [5]. Features are extracted from four informative sources, which include visual-perceptual information (colour, texture and motion), structural information (shot length, shot distribution, shot rhythm, shot clusters duration and saturation), cognitive information (face properties, such as number, positions and dimensions) and aural information (transcribed text, sound characteristics). These features are used for training a parallel neural network system and achieve an accuracy rate up to 95% in distinguish between seven video genres, namely: football, cartoons, music, weather forecast, newscast, talk shows and commercials.

In this study we propose three categories of content descriptors derived at temporal, color and contour-based levels. Compared to existing literature, e.g. MPEG-7 descriptors, they have some advantages. Temporal descriptors (e.g. action) are determined based on the perception of action at different levels (user experiments have been conducted). Color descriptors involve also the perception of colors (transposed from the semantic analysis of artistic animated movies [6]) and color is generically described in terms of statistics of color distribution, elementary hues, color properties (e.g. amount of light colors, cold colors, etc.) and relationship of adjacency and complementarity. Contour descriptors focus not on closed shapes (although difficult to obtain) but propose to describe curved

segments and contour geometry is described individually (e.g. orientation, degree of curvature, symmetry, etc.) or in relation with other neighboring contours (e.g. angular direction, geometric alignment, etc.). The method is transposed from static image indexing, where it has been successfully validated on retrieving tens of semantic concepts, e.g. outdoor, doors/entrances, fruits, people, etc. [7].

The main novel aspect is however the combination of all these parameters for the classification of 7 common genres. Each genre shows some specificity for these parameters (empirically determined), for instance: *animated movies/cartoons* - have particular color properties; *documentaries* - skyline contours are predominant, rhythm is rather slow; *music clips* - high visual rhythm, high action, darker color palette; *news broadcast* - people/face silhouettes are predominant; *commercials* - high rhythm, rather abstract like animated movies; *movies* - homogenous color palette, similar global rhythm, characters/faces occurrence is high, darker colors and *sports* - have few predominant hues, people silhouettes are predominant (see Section 5.1). Exhaustive experimental tests have been conducted on more than 91 hours of video footage and classification is performed using SVM (Support Vector Machines), KNN (K-Nearest Neighbor) and LDA (Linear Discriminant Analysis). Despite the difficulty of this task due to resemblance between several genres (e.g. music clips and commercials, movies and documentaries) the proposed parameters achieve average precision and recall ratios up to 97% and 100%, respectively.

The remainder of this paper is organized as follows: Section 2, Section 3 and Section 4 deal with feature extraction: temporal structure (action), color properties and image structural information (contour), respectively. Experimental results are presented in Section 5 while Section 6 presents the conclusions and discusses future work.

2 Action Descriptors

The first feature set aims to capture the movie temporal structure in terms of *visual rhythm*, *action* and *amount of gradual video transitions*. These parameters are strongly related to movie contents, e.g. music clips have a high visual tempo, documentaries a low action content, commercials a high amount of gradual transitions, etc. The approach is based on some previous work [9]. It is carried out by first performing movie temporal segmentation, which roughly means the detection of video transitions. We detect cuts and two of the most frequent gradual transitions, i.e. fades and dissolves. Cut detection is performed using an adaptation of the histogram-based approach proposed in [10], while fade and dissolve detection are carried out using a pixel-level statistical approach [11] and the analysis of fading-in and fading-out pixels [12], respectively. Further, we determine the following parameters (see also Figure 1):

Rhythm. To capture the movie’s changing tempo, we define first a basic indicator, denoted $\zeta_T(i)$, which represents the relative number of shot changes occurring within the time interval of T seconds, starting from a frame at time

index i ($T = 5s$, experimentally determined). Based on ζ_T , we define the movie rhythm as the movie's average shot change speed, \bar{v}_T , i.e. the average number of shot changes over the time interval T for the entire movie [8], thus:

$$\bar{v}_T = E\{\zeta_T(i)\} = \sum_{t=1}^{T \cdot 25} t \cdot f_{\zeta_T(i)}(t) \quad (1)$$

in which $T \cdot 25$ represents the number of frames of the time window (at 25 fps) and $f_{\zeta_T(i)}$ is the probability density of $\zeta_T(i)$ given by:

$$f_{\zeta_T(i)}(t) = \frac{1}{N_T} \sum_{i \in W_T} \delta(\zeta_T(i) - t) \quad (2)$$

in which N_T is the total number of time windows of size T seconds (defining the set W_T), i is the starting frame of the current analyzed time window and $\delta(t) = 1$ if $t = 0$ and 0 otherwise.

Action. To determine the following parameters, we use the basic assumption that, in general, action content is related to a high frequency of shot changes [13]. We aim at highlighting two opposite situations: video segments with a high action content (hot action) and video segments with low action content [8].

First, at a coarse level, we highlight segments which show high number of shot changes ($\zeta_T > 2.8$), i.e. candidates for hot action, and a reduced number of shot changes ($\zeta_T < 0.71$), i.e. low action. Thresholds were set experimentally after manually analyzing ζ_T values for several representative action segments of each class (adaptation of [8]). To reduce over-segmentation of action segments, we merge neighboring action segments at a time distance below T seconds (the size of the time window). Further, we remove unnoticeable and irrelevant action segments by erasing small action clips less than the analysis time window T . Finally, all action clips containing less than $N_s = 4$ video shots are being removed. Those segments are very likely to be the result of false detections, containing one or several gradual transitions (e.g. a "fade-out" - "fade-in" sequence).

Based on this information, action content is described with two parameters, hot-action ratio (HA) and low-action ratio (LA):

$$HA = \frac{T_{HA}}{T_{total}}, \quad LA = \frac{T_{LA}}{T_{total}} \quad (3)$$

where T_{HA} and T_{LA} represent the total length of hot and low action segments, respectively, and T_{total} is the movie total length.

Gradual Transition Ratio. The last parameter is related to the amount of the gradual transitions used within the movie. We compute the gradual transition ratio (GT):

$$GT = \frac{T_{dissolves} + T_{fade-in} + T_{fade-out}}{T_{total}} \quad (4)$$

where T_x represents the total duration of all the gradual transitions of type x .

3 Color Descriptors

The next feature set aims to capture the movie’s global color contents in terms of statistics of color distribution, elementary hues, color properties and color relationship. This is carried out using an adaptation of the approach proposed in [6]. Prior to the analysis, several pre-processing steps are adopted. To reduce complexity, color features are computed on a summary of the initial video. Each video shot is summarized by retaining only $p = 10\%$ of its frames as a sub-sequence centered with respect to the middle of the shot (experimental tests proved that 10% is enough to preserve a good estimation of color distribution). The retained frames are down-sampled to a lower resolution (e.g. average width around 120 pixels). Finally, true color images are reduced to a more convenient color palette. We have selected the non-dithering 216 color Webmaster palette due to its consistent color wealth and the availability of a color naming system. Color mapping is performed using a minimum $L^*a^*b^*$ Euclidean distance approach applied using a Floyd-Steinberg dithering scheme [14]. The proposed color parameters are determined as follows (see also Figure 1).

Global Weighted Color Histogram captures the movie’s global color distribution. It is computed as the weighted sum of each individual shot average color histogram, thus:

$$h_{GW}(c) = \sum_{i=0}^M \left[\frac{1}{N_i} \sum_{j=0}^{N_i} h_{shot_i}^j(c) \right] \cdot \frac{T_{shot_i}}{T_{total}} \quad (5)$$

where M is the total number of video shots, N_i is the total number of the retained frames from the shot i (i.e. $p = 10\%$), $h_{shot_i}^j()$ is the color histogram of the frame j from shot i , c is a color index from the Webmaster palette and T_{shot_i} is the total length of the shot i . The longer the shot, the more important the contribution of its histogram to the movie’s global histogram. Defined in this way, values of $h_{GW}()$ account for the global color apparition percentage in the movie (values are normalized to 1, i.e. a frequency of occurrence of 100%).

Elementary Color Histogram. The next feature is the elementary color distribution which is computed, thus:

$$h_E(c_e) = \sum_{c=0}^{215} h_{GW}(c) |_{Name(c_e) \subset Name(c)} \quad (6)$$

where c_e is an elementary color from the Webmaster color dictionary (colors are named according to the color’s hue, saturation and intensity), $c_e \in \Gamma_e$ with $\Gamma_e = \{ \text{"Orange", "Red", "Pink", "Magenta", "Violet", "Blue", "Azure", "Cyan", "Teal", "Green", "Spring", "Yellow", "Gray", "White", "Black"} \}$ and $Name()$ returns a color’s name from the palette dictionary. In this way, each available color is projected in $h_E()$ on to its elementary hue, therefore disregarding the saturation and intensity information. This mechanism assures invariance to color fluctuations (e.g. illumination changes).

Color Properties. The next parameters aim at describing, first, color perception by means of light/dark, saturated/non-saturated, warm/cold colors and second, color wealth by quantifying color variation/diversity. Using previously determined histogram information in conjunction with color naming dictionary we define several color ratios.

For instance, light color ratio, P_{light} , which reflects the amount of bright colors in the movie, is computed thus:

$$P_{light} = \sum_{c=0}^{215} h_{GW}(c) |_{W_{light} \subset Name(c)} \quad (7)$$

where c is the index of a color with the property that its name (provided by $Name(c)$) contains one of the words defining brightness, i.e. $W_{light} \in \{ "light", "pale", "white" \}$. Using the same reasoning and keywords specific to each color property, we define:

- dark color ratio, denoted P_{dark} , where $W_{dark} \in \{ "dark", "obscure", "black" \}$;
- hard color ratio, denoted P_{hard} , which reflects the amount of saturated colors. $W_{hard} \in \{ "hard", "faded" \} \cup \Gamma_e$, where Γ_e is the elementary color set (see equation 6, elementary colors are 100% saturated colors);
- weak color ratio, denoted P_{weak} which is opposite to P_{hard} , $W_{weak} \in \{ "weak", "dull" \}$;
- warm color ratio, denoted P_{warm} , which reflects the amount of warm colors; in art, some hues are commonly perceived to exhibit some levels of warmth, namely: "Yellow", "Orange", "Red", "Yellow-Orange", "Red-Orange", "Red-Violet", "Magenta", "Pink" and "Spring";
- cold color ratio, denoted P_{cold} , where "Green", "Blue", "Violet", "Yellow-Green", "Blue-Green", "Blue-Violet", "Teal", "Cyan" and "Azure" are reflecting coldness.

Further, we capture movie color wealth with two parameters. Color variation, P_{var} , which accounts for the amount of significant different colors, is defined thus:

$$P_{var} = \frac{Card\{c | h_{GW}(c) > \tau_{var}\}}{216} \quad (8)$$

where c is a color index, h_{GW} is the global weighted histogram defined in equation 5 and $Card()$ is the cardinal function which returns the size of a data set. We consider a color significant enough for the movie's color distribution if it has a frequency of occurrence of more than 1% (i.e. $\tau_{var} = 0.01$). Color diversity, P_{div} , which reflects the amount of significant different color hues is defined on the elementary color histogram h_E using the same principle.

Color Relationship. The final two parameters are related to the concept of perceptual relation of color in terms of adjacency and complementarity. Hence, P_{adj} reflects the amount of similar perceptual colors in the movie (neighborhood

pairs of colors on a perceptual color wheel, e.g. Itten’s color wheel) and P_{compl} reflects the amount of opposite perceptual color pairs (antipodal).

4 Contour Descriptors

The final descriptor set provides structural information in terms of contours and their relations. Contours are partitioned and represented as described in [7]. Similar to color information, contour information is extracted not from the entire movie but from a summary of the movie. In this case, we aim at retaining around 100 images evenly distributed with respect to video transitions. Retained frames are down-sampled to a lower resolution, whereby maintaining the image’s aspect ratio (e.g. average width around 120 pixels). Contour processing starts with edge detection, which is performed with a Canny edge detector [16], and continues with creation of the local/global space (LG) for each extracted contour, followed by contour partitioning, segment extraction and finally contour description. To capture all relevant structural information, contours are extracted at several spatial scales (e.g. $\sigma=1,2,3,5$, see [16]). At the beginning, descriptors are determined for all four scales but later reduced by keeping only the most symmetric and smooth contours.

Contour Signatures. A contour is iterated with a window (fixed chord: ω) which classifies a segment into three labels: bow, inflexion and straight, and additionally determines the amplitude of the segment. For a given window size, this leads to the ”bowness” $\beta(v)$, inflexion $\tau(v)$ and straightness signature $\gamma(v)$, where v represents the arc length variable. For a range of window sizes, this leads to a set of signatures which describe the LG space, one for bows, $\beta_\omega(v)$, one for inflexions, $\tau_\omega(v)$, and one for straightness, $\gamma_\omega(v)$. The straightness signature is suppressed (γ set to 0) if at the same location a positive bowness value is present in the same or any higher window level ω .

Contour Properties. Further, contours are partitioned at U turns, i.e. sharp curvatures of 180 degrees, which can be located in the bowness space $\beta_\omega(v)$. After application of this rule, any contour appears either as elongated in a coarse sense or as an arc. A contour is thus soft-classified as ”wiggly” and ”arced” by setting a scalar value that expresses the strength of these aspects (w and a respectively; if both values are 0 the contour is straight). From the wiggly contours, long straight and arced segments are extracted, as they could be locally grouped with other neighboring contours. Other geometric aspects that are derived from the LG space are:

- degree of curvature, denoted b ;
- degree of circularity, denoted ζ (for arcs larger than 180 degrees);
- edginess parameter, denoted e , that expresses the sharpness of a curve (L feature or bow);
- symmetry parameter, denoted y , that expresses the ”evenness” of the contour.

In addition to those geometric parameters, a number of "appearance" parameters are extracted. They consist of simple statistics obtained from the luminance values extracted along the contour, such as the contrast (mean, standard deviation; abbreviated c_m , c_s respectively) and the "fuzziness", obtained from the convolution of the image with a blob filter (f_m , f_s , respectively) [7].

Contour Relationship. Contour segments are then grouped, firstly as pairs. For each contour, three neighboring segments are searched (that potentially constitute useful pairs for description): one for each endpoint and one for its center point that forms a potential pair of parallel segments. The selection of appropriate pairs is based on a number of criteria and measures such as the spatial proximity between (proximal) endpoints, the structural similarity of the two segments and the symmetry of their alignment. Selected pairs are then geometrically described by the following dimensions:

- angular direction of the pair, denoted γ_p ;
- distance between the proximal contour end points, denoted d_c ;
- distance between the distal contour end points, denoted d_o ;
- distance between the center (middle) point of each segment, denoted d_m ;
- average segment length, denoted l ;
- symmetry of the two segments, denoted y ;
- degree of bendness of each segment, denoted b_1 and b_2 ;
- structural biases, abbreviated with \hat{s} , that express to what degree the pair alignment is a L feature (\hat{s}_L), T feature (\hat{s}_T) or a "closed" feature (two curved segments facing each other as '()', $\hat{s}_()$).

In summary, at the image level, structural information is represented in a statistical manner using histograms. For each descriptor parameter, a 10-bin histogram is generated. The histograms are then concatenated to form a single descriptor vector. At movie level, feature vectors are averaged, forming so the structure signature of the movie.

5 Experimental Results

To test the discriminative power of the proposed parameters in video genre classification, we have selected 7 of the most common genres, namely: *animated movies*, *commercials*, *documentaries*, *movies*, *music videos*, *news broadcast* and *sports*. Each genre is represented with 30 sequences recorded from several TV programmes, summing up to 210 sequences and more than 91 hours of video footage, thus: 20h30min of animated movies (long, short clips and series), 15min of commercials, 22h of documentaries (wildlife, ocean, cities and history), 21h57min of movies (long, episodes and sitcom), 2h30min of music (pop, rock and dance video clips), 22h of news broadcast and 1h55min of sports (mainly soccer).

5.1 Parameter Examples

In Figure 1 and 2 we present average color (see Section 3), action (see Section 2) and contour (see Section 4) feature vectors for each of the 7 genres.

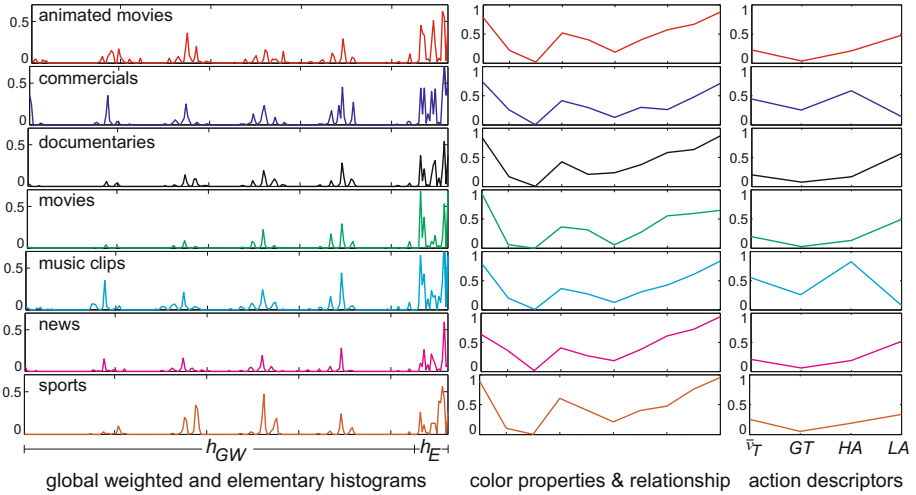


Fig. 1. Average color-action feature vectors for each genre

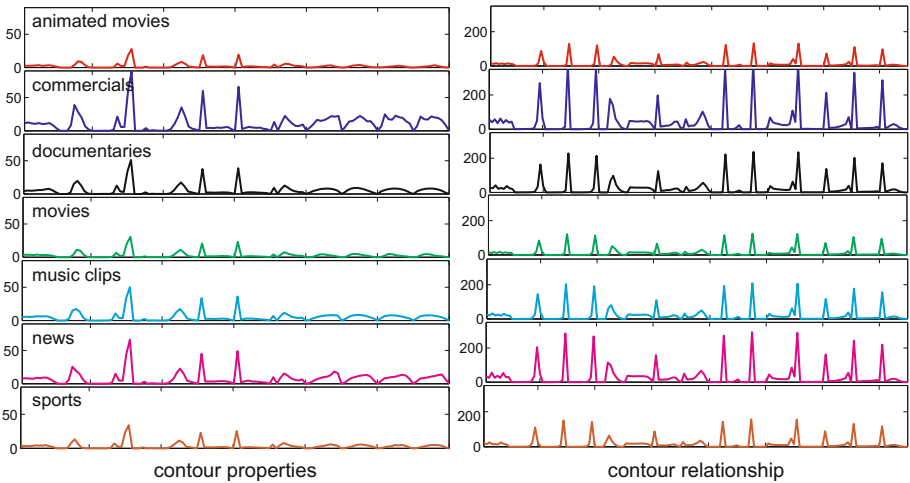


Fig. 2. Average contour feature vectors for each genre

In average, the proposed parameters show a different comportment for each genre. For instance, commercials and music clips have a high visual rhythm and action content (see \bar{v}_T and HA in Figure 1), animated movies have a different color pattern (more colors are being used, see h_{CW}) and most of the hues are used in important amounts (see h_E), movies and documentaries tend to have a reduced action content, sports have a predominant hue (see the predominant peak in h_E), commercials show an important symmetry of contours (see contour relationship in Figure 2), and so on. Discriminant power of the features is evidenced however in the classification task below.

5.2 The Classification Approach

Genre retrieval is carried out with a binary classification approach, i.e. one genre at a time vs. all others. We test several supervised classification methods, namely: the K-Nearest Neighbors algorithm (KNN, with $k=1$, cosine distance and majority rule), Support Vector Machines (SVM, with a linear kernel) and Linear Discriminant Analysis (LDA, with linear discriminant function applied on a PCA-reduced feature space) [15]. The method parameters were set to optimal values for this scenario after several tests.

As the choice of the training set may distort the accuracy of the results, we have adopted an exhaustive testing, i.e. training sequences are selected randomly and each classification is repeated over 1000 times in order to extract all possible combinations. Also, tests were performed for different amounts of training data, as depicted in Table 1.

Table 1. Training sets from 210 test sequences

% training data	10%	20%	30%	40%	50%	60%	70%
total nb. of training sequences	21	42	63	84	105	126	147
(from which) # of current genre:	3	6	9	12	15	18	21
total nb. of test sequences	189	168	147	126	105	84	63
(from which) # of current genre	27	24	21	18	15	12	9

To assess performance we adopt several strategies. First, we evaluate average precision (P) and recall (R) ratios for each target class, thus:

$$P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}, \quad R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \quad (9)$$

where \overline{TP} , \overline{FP} and \overline{FN} represent the *average* number of good detections (true positives), false detections (false positives) and non detections (false negatives), respectively. Secondly, to provide a global measure of performance we compute the average correct detection ratio, denoted \overline{CD} , and F_{score} ratio, thus:

$$\overline{CD} = \frac{\overline{N_{GD}}}{N_{total}}, \quad F_{score} = 2 \cdot \frac{P \cdot R}{P + R} \quad (10)$$

were $\overline{N_{GD}}$ is the average number of good classifications (for both classes, target genre and others) and $N_{total} = 210$ is the number of test sequences.

5.3 Discussion on Precision and Recall

Despite the strong semantic resemblance between different genres, the proposed parameters achieve good classification results. Figure 3 depicts the precision vs. recall curves for different amounts of training data (see Table II) and an average Fscore ($\overline{F_{score}^g}$) over all genres. For all genres we achieve detection ratios above 80%, while for some particular genres detection ratios are close to 100%.

Due to the similarity of the content, the weakest classification performance is obtained for music and commercials, thus:

- for *music* the highest accuracy is obtained with LDA using color-action ($R = 81\%$) and the lowest false positive rate with KNN using contour-color-action ($P = 75\%$);
- for *commercials* the highest accuracy is obtained with LDA using color-action ($R = 72\%$) and the lowest false positive rate with KNN using contour-color-action ($P = 89\%$).

The high diversity of video material (short clips, long movies, episodes, cartoons, artistic animated movies) situates *animated movies* on an average classification performance, thus the highest accuracy is obtained with LDA using contour-color-action ($R = 84\%$) while the lowest false positives rate with SVM using contour-color-action ($P = 88\%$).

A relatively high classification accuracy is obtained for genres which show at some level a certain homogeneity in structure and content, namely: documentaries, movies, news and sports:

- for *documentaries* the highest accuracy and the lowest false positives rates are both obtained with KNN using contour-color-action ($R = 96\%$, $P = 80\%$);
- for *movies* the highest accuracy is obtained with LDA using color-action ($R = 95\%$) while the lowest false positives rate is obtained with SVM using contour-color-action ($P = 87\%$);
- for *news* the highest accuracy is obtained with KNN using contour-color-action ($R = 100\%$), as well as with KNN using contour and LDA using color-action (however, precision is lower for the last two), while the lowest false positives rate is obtained with SVM using color-action ($P = 85\%$);
- for *sports* the highest accuracy is obtained with LDA using color-action ($R = 94\%$) while the lowest false positives rate is obtained with KNN using contour-color-action ($P = 97\%$).

In general, detection ratios increase with the amount of training data (see Figure 3). Also, one may observe that the best performance tends to be achieved with the fusion of contour, color and action parameters.

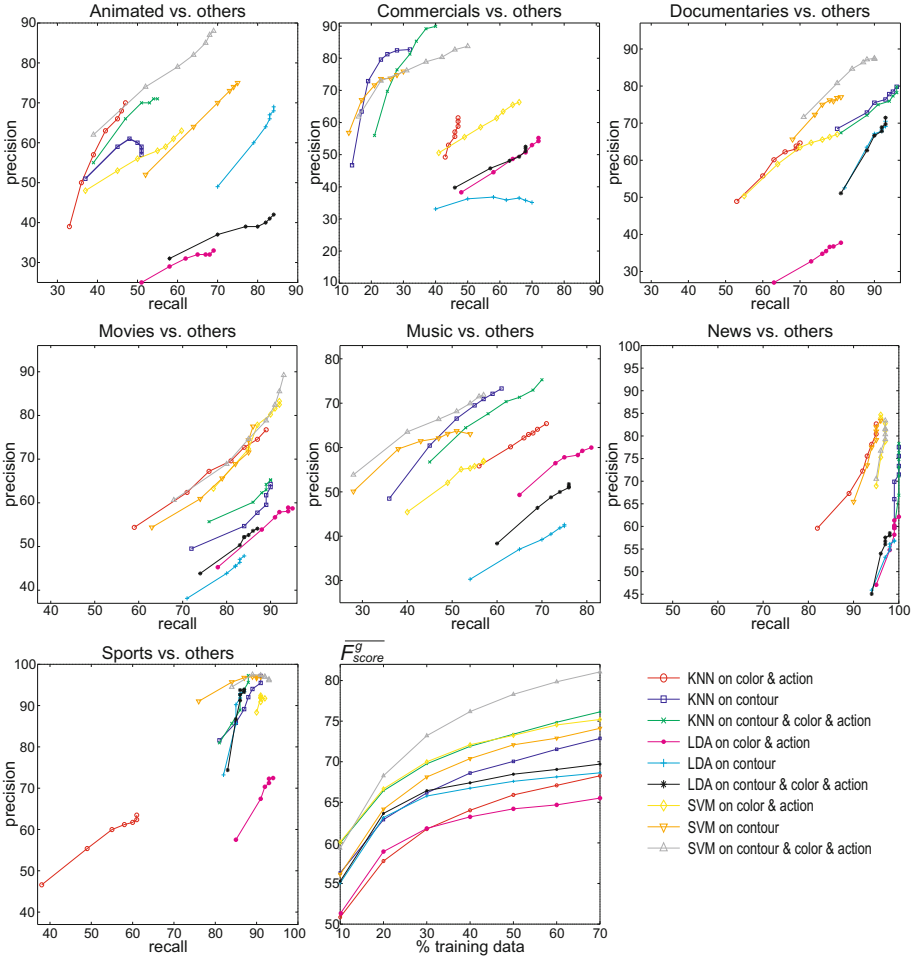


Fig. 3. Precision vs. recall curves for different runs and amounts of training data (% of training is increasing along the curves). \overline{F}_{score}^g represents the average Fscore measure achieved for all genres.

5.4 Discussion on Global Fscore and Correct Detection Ratio

To assess overall performance, we use an average Fscore measure (\overline{F}_{score}^g , see Figure 3) and the average correct detection ratio (\overline{CD} , see Figure 4) which are computed over all genres (we use averaging as each genre is represented with equal number of sequences). Based on this information, the most powerful approach proves to be, again, the combination of all three feature classes, i.e. contour, color and action.

It provides the best result with SVM and KNN classifiers, namely: SVM - $\overline{F}_{score}^g = 81.06\%$, $\overline{CD} = 94.92\%$, followed by KNN - $\overline{F}_{score}^g = 76.14\%$, $\overline{CD} = 93.34\%$. Also, SVM provides the highest accuracy for the smallest training set

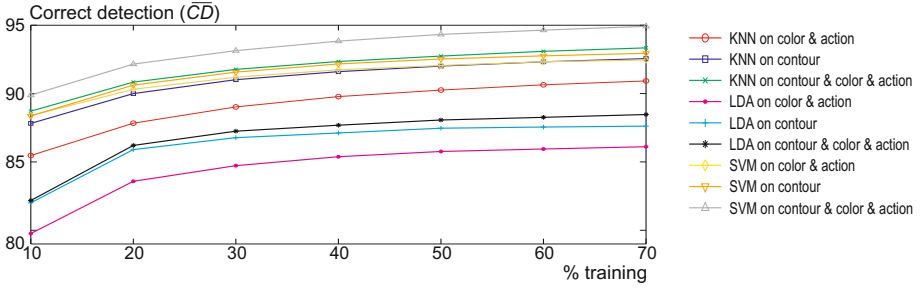


Fig. 4. Average correct detection ratio for all genres (\overline{CD} , the x axis represents the amount of training data, see Table II)

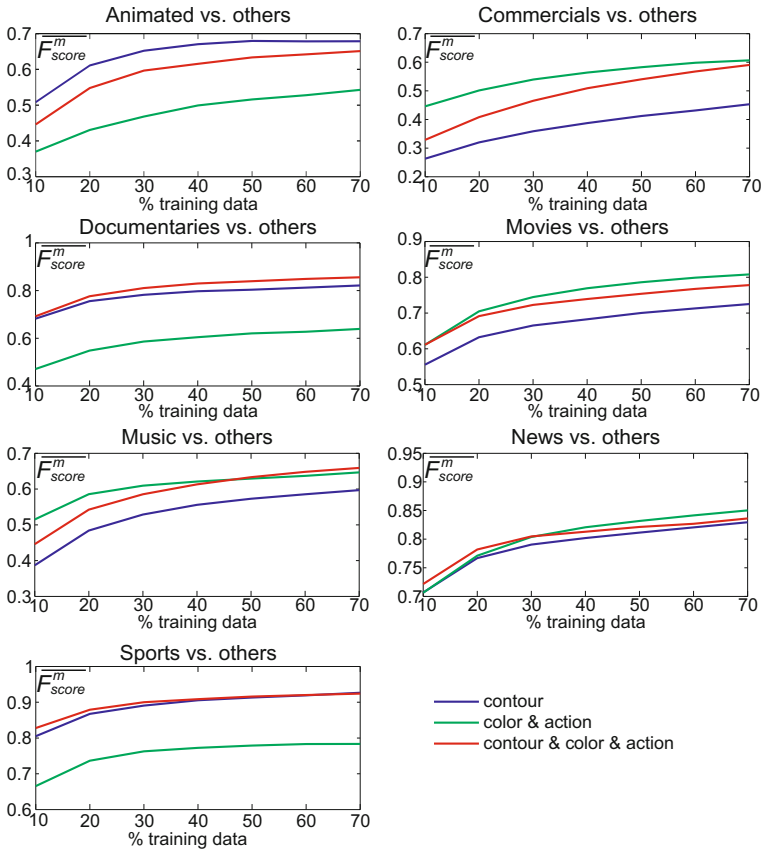


Fig. 5. Average Fscore measure per feature set and for all methods (KNN + SVM + LDA, F_{score}^m)

(see Figure 4), e.g. $\overline{CD} = 89.89\%$ (training 10%, see Table II) or $\overline{CD} = 92.15\%$ (training 20%).

The final test was to determine the overall discriminant power of each feature set. For that, we compute an average Fscore for all three methods (KNN + SVM + LDA), denoted $\overline{F_{score}^m}$. The results are presented in Figure 5. Although this tends to be a relatively subjective evaluation, being related to the performance of the selected classification methods, we obtain some interesting results. Contour parameters, compared to color-action parameters, preserve their efficiency in retrieving specific object shapes, as proved for static images 7. They provide the highest score for documentaries (skyline contours are frequent, $\overline{F_{score}^m} = 82.12\%$), sports (people silhouettes are frequent, $\overline{F_{score}^m} = 92.42\%$) and good results for news (anchorman bust silhouette and people silhouette are frequent, $\overline{F_{score}^m} = 82.96\%$). On the other hand, compared to contours, color-action features perform better for music ($\overline{F_{score}^m} = 64.66\%$), commercials ($\overline{F_{score}^m} = 60.66\%$), movies ($\overline{F_{score}^m} = 80.8\%$) and news ($\overline{F_{score}^m} = 85\%$) which can be assigned to the specific rhythm and color diversity of each genre. However, these preliminary results show that in general each genre distinguishes itself from the others by a specific set of descriptors.

6 Conclusions and Future Work

We addressed the issue of automatic classification of video genres and proposed several types of content descriptors, namely: temporal, color and contour structural parameters. These descriptors are used to retrieve 7 common video genres (tests were performed on 91 hours of video footage in total).

At individual level, all genres achieve precision and recall ratios above 80%, while some genres achieve even higher detection ratios, close to 100%. Overall, for all genres, the combination of all descriptors, i.e. contour-color-action, provided the highest accuracy and achieves an average Fscore ratio above 80%, while the average correct detection ratio is above 94%. Finally, at feature level, average Fscore ratios are up to 92%.

One limitation of this approach is in its computational complexity. Color and action descriptors rely mainly on temporal segmentation (cuts, fades, dissolves) which can be time consuming, while contours are extracted at different levels and re-refined after extracting the descriptors. Despite the fact that all processing, i.e. contour-color-action, takes less than half the sequence duration, to be integrated with a real indexing system, hardware acceleration/optimization is required.

However, these represent some of our preliminary work. Future work will include reducing data redundancy, addressing higher semantic levels of description (e.g. exploiting concept detection) as well as extending tests on a larger scale database.

Acknowledgments. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

References

1. Smeaton, A.F., Over, P., Kraaij, W.: High-Level Feature Detection from Video in TRECVID: a 5-Year Retrospective of Achievements, *Multimedia Content Analysis. In: Theory and Applications*, pp. 151–174. Springer, Berlin (2009); ISBN 978-0-387-76567-9
2. Brezeale, D., Cook, D.J.: Automatic Video Classification: A Survey of the Literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38(3), 416–430 (2008)
3. Roach, M.J., Mason, J.S.D.: Video Genre Classification using Dynamics. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, Utah, USA, pp. 1557–1560 (2001)
4. Yuan, X., Lai, W., Mei, T., Hua, X.S., Wu, X.Q., Li, S.: Automatic video genre categorization using hierarchical SVM. In: *IEEE International Conference on Image Processing*, Atlanta, USA, pp. 2905–2908 (2006)
5. Montagnuolo, M., Messina, A.: Parallel Neural Networks for Multimodal Video Genre Classification. *Multimedia Tools and Applications* 41(1), 125–159 (2009)
6. Ionescu, B., Coquin, D., Lambert, P., Buzuloiu, V.: A Fuzzy Color-Based Approach for Understanding Animated Movies Content in the Indexing Task. *Eurasip Journal on Image and Video Processing* (2008), doi:10.1155/2008/849625
7. Rasche, C.: An Approach to the Parameterization of Structure for Fast Categorization. *International Journal of Computer Vision* 87(3), 337–356 (2010)
8. Ionescu, B., Pacureanu, A., Lambert, P., Vertan, C.: Highlighting Action Content in Animated Movies. In: *IEEE ISSCS - International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 9–10 (2009)
9. Ionescu, B., Coquin, D., Lambert, P., Buzuloiu, V.: Semantic Characterization of Animation Movies Based on Fuzzy Action and Color Information. In: Marchand-Maillet, S., et al. (eds.) *AMR 2006. LNCS*, vol. 4398, pp. 119–135. Springer, Heidelberg (2007)
10. Ionescu, B., Buzuloiu, V., Lambert, P., Coquin, D.: Improved Cut Detection for the Segmentation of Animation Movies. In: *IEEE International Conference on Acoustic, Speech and Signal Processing*, Toulouse, France (2006)
11. Fernando, W.A.C., Canagarajah, C.N., Bull, D.R.: Fade and Dissolve Detection in Uncompressed and Compressed Video Sequence. In: *IEEE International Conference on Image Processing*, Kobe, Japan, pp. 299–303 (1999)
12. Su, C.-W., Liao, H.-Y.M., Tyan, H.-R., Fan, K.-C., Chen, L.-H.: A Motion-Tolerant Dissolve Detection Algorithm. *IEEE Transactions on Multimedia* 7(6), 1106–1113 (2005)
13. Chen, H.W., Kuo, J.-H., Chu, W.-T., Wu, J.-L.: Action Movies Segmentation and Summarization based on Tempo Analysis. In: *ACM International Workshop on Multimedia Information Retrieval*, New York, pp. 251–258 (2004)
14. Floyd, R.W., Steinberg, L.: An Adaptive Algorithm for Spatial Gray Scale. In: *Proc. SID Int. Symp. Digest of Technical Papers*, pp. 36–37 (1975)
15. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005); ISBN 0-12-088407-0
16. Canny, J.: A Computational Approach To Edge Detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)

Differences in Video Search Behavior between Novices and Archivists

Henning Rode, Theodora Tsikrika, and Arjen P. de Vries

Centrum Wiskunde & Informatica,
Science Park 123, Amsterdam, The Netherlands
henning.rode@cwi.nl, {theodora.tsikrika,arjen}@acm.org

Abstract. Improving users' interactions with a video retrieval system requires the examination of the search behavior of real users. This paper presents a study that examines and compares the video search behavior of professional archivists and novice users. The comparison focuses on the use and effectiveness of different state-of-the-art video search methods offered by the VITALAS retrieval system, and also on the behavior of the two user groups during their interactions with the retrieval results. We conducted our experiments in the context of TRECVID's 2009 interactive search task, using the provided collection and topics for our evaluation. The findings are based on a qualitative questionnaire analysis and a quantitative examination of the logged user actions on the search interface. The experimental results indicate that today's visual search techniques have improved in effectiveness, confirming a trend found in previous user studies. To our surprise, professional archivists used visual concept search in many of their searches. Queries containing visual concepts were more effective, resulting in more relevant shots found than the alternative methods. Overall, we conclude that professional archivists are more focused on recall when carrying out their search tasks and are better at reflecting on their own search performance.

Keywords: video retrieval, user study, TRECVID, professional archivists, novice users.

1 Introduction

VITALAS (<http://vitalas.ercim.org/>) is an EU-funded Integrated Project that aimed at the development of a system capable of large-scale indexing and retrieval of video and images, specifically targeted towards multimedia professionals and archivists. The VITALAS video retrieval system integrates different state-of-the-art search methods into a single user interface that supports search on (the combination of) multiple modalities. This paper reports on two series of experiments, the first carried out within the context of TRECVID's 2009 interactive search task and a later independent follow-up experiment that used the same test collection. The aim of both experiments has been to analyze and compare the use and user-perceived effectiveness of the available search functionalities. We examined how different types of users interact with the system by

comparing the behavior of professionals and novice users. Archivists employed by three large-scale broadcasting archives (located in France, Germany and The Netherlands) participated as professional users in our study, while novice users were represented by people who are not professionally concerned with video retrieval. We performed a quantitative and qualitative evaluation of users' search behavior by using questionnaires and analyzing the search logs.

The remainder of this paper is structured as follows. Section 2 presents related user studies that examine the behavior of users interacting with video retrieval systems. Section 3 describes the VITALAS video retrieval system, its search components and user interface, and the used test collection. Section 4 details the setup and execution of the user tests. Section 5 presents the results of the experiments and tries to point out differences and common findings with the above mentioned related user studies. Section 6 concludes this work.

2 Related Work

Several previous works have studied multimedia retrieval in an interactive setting. For example, Westman et al. [15] conducted a comparison of professional and non-professional users of an image retrieval system. In their study, editors of the magazine industry participated as professionals and carried out retrieval tasks representative of their daily work. Hollink et al. [11] analyzed the search patterns arising from the use of their news video retrieval system by non-professional users; they distinguished search behavior with respect to different search task categories and contrasted searching for specific known items or entities with more generic search tasks.

The majority of user studies that have examined the video searching behavior of users interacting with video retrieval systems have though been carried out in the context of TRECVID's interactive search task, see e.g., [14,13] and the studies discussed below. In the TRECVID framework, interactive search is defined relative to a given video collection, a common shot reference for this collection, and a multimedia statement of an information need referred to as *topic*, with the aim to return a ranked list of up to N ($= 1000$) shots which best satisfy the need. Users participating in the task have no prior knowledge of the search test collection or topics. A review of the video retrieval approaches investigated during the first three years of TRECVID (2001-2003) revealed that interactive approaches consistently and substantially outperformed all non-interactive approaches [9]. In this period, "query-by-text", rather than "query-by-image" and "query-by-concept" accounted for most of the successful interactions, particularly among novices [5].

Our study is most similar in goals and research methods used to the research carried out by the Informedia research group (during several years of TRECVID), so we will discuss their work [3,4,2,6] in more detail. Similar to our analysis, these studies combine questionnaires and search log analysis in order to examine user behavior. They compare in depth the effectiveness of different search methods and querying paradigms, looking separately into the performance of "expert" users and "novices". Here, the authors distinguish experts

from novices on the basis of three types of knowledge that only their experts possess [2]: (1) experts have been working with the research group that develops the video retrieval system to be evaluated and thus have a better understanding of the various automated video processing techniques; (2) the expert has used the tested video retrieval system prior to the user study with the TRECVID data, perhaps even contributing to its development, and therefore knows the system operation better than the users who first interact with it during the user tests; and (3) the expert is familiar with TRECVID evaluation, e.g., the emphasis on shot-based retrieval and use of mean average precision as a key metric [4].

In the TRECVID 2004 user studies conducted by Informedia [3], it was observed that expert users were more willing to use image-based and concept-based search, whereas novice users mostly engaged in “query-by-text” interactions. The much better performance by the experts in that year’s search tasks was attributed to the use of all three querying strategies and motivated the development of a video retrieval system that would enable increased use of the different search methods. The following year, the user interface of their system was redesigned so as to promote the use of non-text based video access mechanisms [4], achieving the desired effect of increasing the use of image-based and concept-based searches by novice users, rather than exclusively relying on text search. In TRECVID 2006 [2,6], the “query-by-text” strategy was even less dominant and the employment of concept-based queries was observed to be an effective strategy for reducing the result shot space. In both 2005 and 2006, the Informedia team recruited government intelligence analysts to participate in the experiments.

While these studies share the overall aims and research method with the work discussed here, there are notable differences in the setup which justify the effort put into our own experiments. First, the TRECVID collection has changed considerably, changing its contents from the broadcast news domain to a more heterogenous collection that includes documentaries and educational programs. TRECVID’s 2009 search topics concentrate on generic search tasks only and do not include known item or entity searches. Also, previous user studies were conducted more than two years ago, a time span in which the state-of-the-art video retrieval methods, and in particular concept-based search, have improved, as it is evidenced by the results obtained in fully automatic machine-only evaluations. Finally, we believe that the archivists participating in our study represent a different type of professional users that come with different experiences and needs, even when compared to intelligence analysts.

3 Video Retrieval System and Test Collection

The VITALAS video retrieval system used in this study allows users to make use of the following search functionalities: (1) keyword search, (2) concept search, (3) visual similarity search, (4) fused search (any combination of the above), and (5) concept suggestions.

¹ According to this definition, our professional archivists would not be considered to be expert users!

The text search component allows search on the text output generated from automatic speech recognition (ASR) on the video material. It provides common full-text search functionalities, such as keyword and phrase search, and returns a ranked list of shots. The concept search retrieves shots based on automatically detected concepts. Similarly to keyword queries, the user can search for any combination of concepts and the system ranks the shots in the collection based on the estimated combined relevance. Visual similarity search completes the query functionalities offered. Once users have found one or more relevant examples, they can use them to search for shots in order to find visually similar keyframes. Similarity search thus enables the retrieval of shots based on visual features, without being bound to the predefined set of concepts. Users are further supported by a concept suggestion service. Whenever a user issues a text search, the service returns concept suggestions related to the submitted query. In this way, users are made aware of automatically detected concepts that might be useful for refining or expanding their queries.

Figure 1 shows the main query interface as well as the *result view*. The top text field allows users to enter keyword and concept queries. The retrieved shots are shown by a thumbnail keyframe in the mosaic result view below. Each thumbnail can be added to a lightbox that is used for gathering possibly relevant shots for the search topic. Clicking on thumbnails opens a *zoom view* showing a single keyframe in a higher resolution as presented in the figure. The zoom view also enables to enter a *detailed view* in order to play the shot (Figure 2). The thumbnails can also be used to issue a visual similarity search. The detailed view allows users to play the shot and to jump to any other shot in the video. (At the time of the experiments, the test prototype of the system did unfortunately not allow to add other shots from the detailed view to the lightbox.) Hotkeys allow a more efficient user handling for the experienced user. The “suggestion bar” shows concept suggestions derived from the previous query’s results, that upon the user’s click are added to the current query.



Fig. 1. VITALAS video retrieval system user interface: zoom view

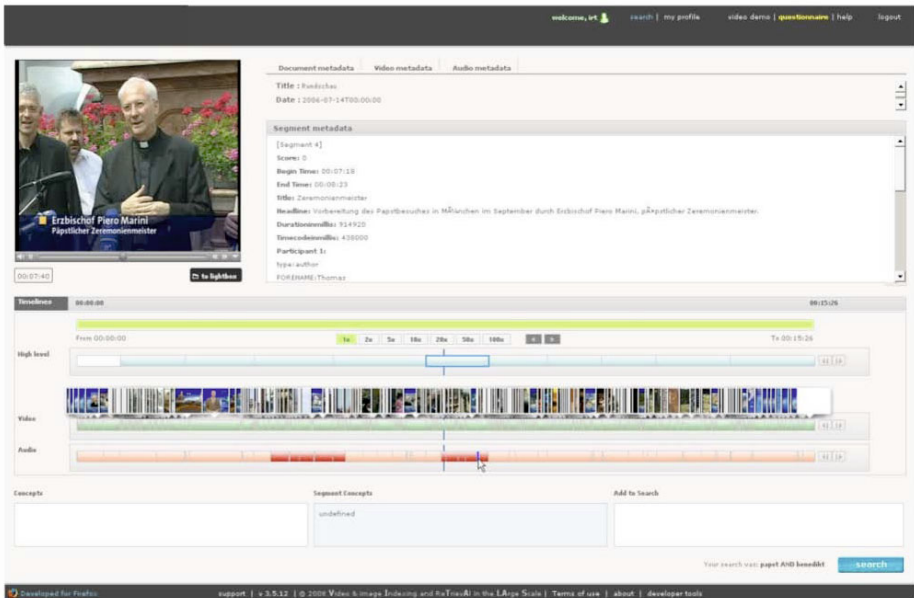


Fig. 2. VITALAS video retrieval system user interface: detailed view

The TRECVID test collection for the search task in 2009 consists of 280 hours of Dutch video data of news magazines, science news, news reports, documentaries, educational programming, and archival video from the Dutch national broadcasting archive. For the keyword search component, we used a machine-translated version of the ASR output provided by the LIMSI system [8]. (We used the Google machine translation services² to automatically translate the speech transcripts to English text.) The collection is also annotated with 64 different concepts detected automatically. We used the concept detector output for the 20 TRECVID 2009 concepts generated from CERTH-ITI’s concept detector described in [7], extended with the detector output for an additional 44 concepts publicly provided by the MediaMil³ [13]. Machine-translated transcripts and concepts were indexed and made searchable using the open source PF/Tijah retrieval system [10]. For the visual similarity search, features of all keyframes were extracted and indexed by Maestro, INRIA’s visual search component described in [1].

4 User Test Setup

The user tests were split in two phases, one taking place in the context of TRECVID’s 2009 interactive search task and one in a later independent user study that also employed the same test collection. The system deployed in the

² <http://code.google.com/p/google-api-translate-java>

³ <http://www.science.uva.nl/research/mediamill/>

second test phase was an upgraded version of the TRECVID prototype. Due to the system differences, albeit rather minor as explained below, the two test phases are examined separately without conflating the user groups that participated in each of them. It should be noted that none of the test users participated in both test phases.

4.1 Test Phase 1

The first test phase was conducted by following the schedule of the TRECVID 2009 interactive search task. We recruited a total of 10 users to participate in our experiments: 4 professional archivists employed in institutes hosting large archives of public broadcasting (3 from France and 1 from The Netherlands) and 6 non-professional users. None of the users had been involved in the design or implementation of the VITALAS video retrieval system; therefore, in order to gain some familiarity with the system interface and supported functionalities, all users completed a training session prior to their main search sessions. Given their familiarity with the daily use of thesauri for searching their own archive, it seemed realistic to assume that archivists would be familiar with the system's recognized concepts. Hence, we additionally provided professional users with a list of the available concepts. Both user groups could use the system's concept suggestions to modify their searches.

Each user was required to complete 12 of the 24 TRECVID 2009 topics, assigned to them based on a latin squares arrangement. (The order of the topics was not randomized, so that in principle a learning effect across user groups could be observed.) Each user could spend a maximum of 10 minutes on each topic before proceeding to the next one. Users were instructed to save those shots that they considered to be relevant to the topic in question. However, the instructions did not emphasize that they should find as many relevant shots as possible, which resulted in only a few saved shots per topic (about 9 on average). This indicates that the users may have focused only on the shots they considered highly relevant, instead of fulfilling the actual TRECVID task of collecting all relevant shots. The system logged all user interactions, including the submitted queries, the shots viewed, and the shots selected as relevant (i.e., added to the lightbox).

Apart from collecting the raw user interaction data, users were asked to fill in questionnaires at different stages of the experiment: (i) an *entry questionnaire* for collecting background information on the searchers, (ii) a *search questionnaire* provided after each topic to ask users about their perception of the just performed search, and (iii) an *exit questionnaire* which asked for an overall evaluation of the VITALAS system and the functionalities it offers.

4.2 Test Phase 2

A second phase of the user tests was conducted half a year later, when a different group of professional archivists was available. Meanwhile, the retrieval system had undergone a few changes - an unavoidable drawback of carrying out this

research using an integrated system consisting of components that are being developed independently.

The largest difference between the two system variants is that users in the second phase could see zoom views of keyframes by mouse-over, instead of requiring a click on the thumbnail representation in the result overview. This modification eased the interaction with the system and consequently increased the number of zoom actions in the searchlogs, rendering though, at the same time, each zoom interaction a less conscious user decision. A second, rather unfortunate difference between the two system variants is that the concept suggestion service was not available during this second test period. As a resolution, all users (professionals as well as novices) were provided with the list of automatically detected concepts.

Apart from these system changes, topic assignment had to be modified as well, as the second group of users was more time constrained with respect to total availability for the entire test. We decided to stick to the maximum duration 10 minutes per topic and therefore had to reduce the number of topics. We had 4 archivists (located in Germany), each working on 8 TRECVID 2009 topics, and 5 novice users completing 4 topics each. Similar to the first test phase, all users were asked to fill in questionnaires, and all search interactions were logged by the system.

5 Results

This section presents results from evaluating the gathered questionnaire data and complements the qualitative information from the questionnaires with a quantitative analysis based on the interaction data collected during the experiments.

5.1 Questionnaire-Based Analysis

Table [11](#) presents the users' own perception of effectiveness and ease of use of the available search methods by averaging the assessments collected in their exit questionnaires.

Archivists indicated no difficulties in using the new, visual access methods, although such methods were not available in their usual daily work practice. Novice users expressed a preference for the text search over the other search methods, with respect to ease-of-use. Both groups value the concept search highest with respect to the perceived search effectiveness, followed by fused and text queries. In general, novice users tend to give higher grades with respect to the perceived effectiveness of the different search methods. It should be remarked that the table only shows averages. Differences between individual users are rather large, resulting in a high variance of individual values, in particular for the novices (in line with findings reported by [16](#), further discussed in the conclusions of this paper).

The search questionnaires inquired after each topic assignment whether the user felt they had had sufficient time to work on the topic. While novice users

Table 1. Users' perception of search methods

		archivists		novices	
test phase	search method	ease	effectiveness	ease	effectiveness
phase 1	text	4.25	3.00	4.00	3.16
	concept	4.25	3.50	3.00	3.50
	similarity	4.25	2.00	3.16	3.00
	fused	4.25	2.75	3.66	3.83
phase 2	text	3.66	2.25	4.60	3.00
	concept	4.00	2.75	3.40	3.50
	similarity	4.25	1.25	4.00	3.20
	fused	3.00	2.25	3.20	3.00
questionnaire scale: 1 – 5					

judged the given 10 minutes as sufficient in all cases, independent from the actual perceived search success, professionals would have liked to continue their search interactions for a number of topics, particularly in those cases where they expected to find more relevant results. We also observe for professional users a higher correlation between the self-judged completeness of a search and the satisfaction with search time. The Informedia study [2] does not compare novices and professionals in this respect, but our findings for the professionals confirm their results: the intelligence analysts in their studies were also willing to search longer, even when the satisfaction with the found results remained low. Novices expressed a higher confidence in the quality of their search results, although their self-assessment of success will not be confirmed when we later take a look at the effectiveness measured by the TRECVID assessments.

5.2 Search-Log Analysis

The analysis of interaction data concentrated on the following four aspects of search behavior: (1) the use of the available search methods, (2) the effectiveness of these search methods, (3) the agreement between TRECVID assessors and our test users, and (4) users' interaction with the displayed results.

Use of the Search Methods. Figure 3 shows how often users made use of each search method. More precisely, we count the number of queries that contain at least one text search, concept search, similarity search, or fusion predicate. Therefore, a single query can count for multiple methods at the same time. We observe that the text search was employed most often, followed by concept search, fused search, and similarity search. Looking only at the first query that a user issued when starting with a new topic (referred to as *entry* search), we can see that text and concept search are most often used as entry queries. Similarity searches could not be used to start a new topic since the system only allows to start a similarity search from already found keyframes.

During the first test phase the distribution of search methods stayed similar among the user groups. However, we can observe a clear difference between professional and novice users in the second test phase. Especially the concept

search was more widely employed by the professionals. This could be partly attributed to the fact that concept suggestions were not provided during the second test phase. Apparently, professionals were nevertheless willing to use concepts by consulting the provided list of available concepts, while novice users concentrated on other search methods. We found that the differences in the use of the concept search are statistically significant (with p -value of 0.024 according to the t -test) despite the small test group sizes. Moreover, the difference stays significant even if we restrict the comparison to the second test phase, where both test groups were provided with a concept list.

A comparison with the Informedia study [2] shows a number of similarities: (1) Their professionals also made more use of the visual search methods than the novices. (2) Among the visual search methods, the novices slightly prefer the similarity search, while professionals use the concept search more often. The tendency for more visual searches is also confirmed when taking into account the slightly older study of Hollink et al. [11]. Of course, we should be careful drawing firm conclusions from comparing these studies with ours, as those were carried out on a different data set. Both previous studies report an average number of issued queries per topic of approximately 7 query reformulations, with a lower proportion of visual search methods applied. While we cannot tell why our user groups issue visual search methods more often, we identified three possible (partial) explanations: the improvements observed in automatic benchmarks of visual search techniques, a relatively low effectiveness of searching machine-translated speech transcripts, and the larger heterogeneity in the collection.

Effectiveness of the Search Methods. With respect to the effectiveness of the different search methods, the analysis can be performed by a system-oriented or a user-oriented perspective. A system-oriented analysis examines the number of results retrieved by the system that are also judged as relevant by the TRECVID assessors. A user-oriented analysis considers the number of results added to the lightbox by the test users and thus considered as relevant by them. Table 2 presents the results of these two analyses. Concept search is by far the most effective search strategy from the system perspective. The high difference may be explained by the fact that most TRECVID topics have well matching available concepts. From a user perspective, the differences are considerably smaller, but still a search containing concepts leads on average to a higher number of shots added to the lightbox. Hence, we can state that the measured system level effectiveness is confirmed by the user experience. The results also roughly correspond to the user experiences expressed in the questionnaires shown in Table 1. Hence, the users were able to estimate the effectiveness of the different methods. We cannot explain why testers of the second phase selected in all cases less shots to add to the lightbox, but at least the results of the two test phases confirm each other with respect to the order of effectiveness.

Agreement with TRECVID Assessors. We also evaluated the agreement between our test users and the TRECVID assessors. We estimate the agreement

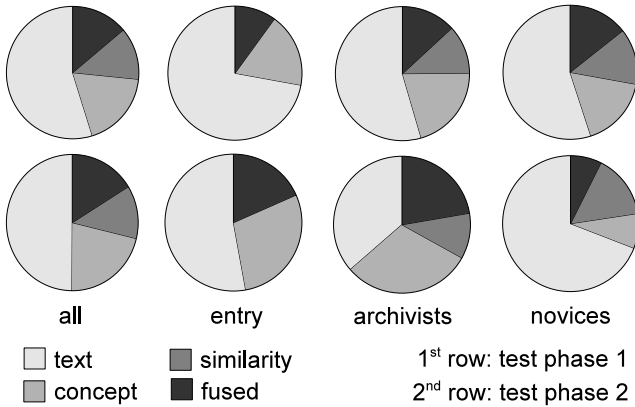


Fig. 3. Proportions of the different search methods

Table 2. System vs. user effectiveness

search type	test phase 1		test phase 2	
	relevant retrieved	added to lightbox	relevant retrieved	added to lightbox
text	2.76	0.98	3.53	0.74
concept	13.47	2.69	12.48	1.04
similarity	4.32	1.07	3.24	0.63
fused	7.88	1.79	7.2	0.70

Table 3. Agreement between users and TRECVID assessors. The table shows the percentage of added, resp. rejected shots annotated as (ir-)relevant by the assessors.

	test phase 1		test phase 2	
	relevant	irrelevant	relevant	irrelevant
added	52%	48%	75%	25%
rejected	43%	57%	32%	68%

level here using two measurements: (1) we look at the shots selected by the user and *added* to the lightbox, and compute the ratio of agreement with the assessors, i.e., the shots the assessors have marked as *relevant* or *irrelevant*⁴, and (2) we look at the shots the users have examined in more detail by zooming in, but have not added to the lightbox; we interpret the latter shots as consciously *rejected* by the users. Table 3 displays the results of this analysis. The first test phase revealed a high level of disagreement, which can however be attributed to a rather high number of user judged shots that was unjudged (i.e., were not part of the assessment pool). In the second phase we observe a considerably higher agreement of about 75% for shots added to the lightbox, and 68% for the users' rejections. Such agreement level seems more reasonable, and comparable to the agreement measured in the Informedia user study [2].

⁴ The shots not explicitly judged by the assessors are considered as irrelevant.

Comparison Among Users. We also examined whether users who had been assigned the same topic found the same or different shots. Although each topic was assigned to a total of five users, the proportion of common shots found by more than one user within all added shots for a topic is only 17%. Hence, by far most of the added shots for a given topic are unique among our users. This could be due to the fact that our users added on average only 9 shots per topic to the task search results. This number is even lower than the 30 shots per topic found by the Informedia users [2] (who could however search for 15 minutes each). We agree therefore with Christel’s assumption that real users are not willing to search for arbitrary many relevant results; after finding several relevant shots, users regard the topic as finished.

Result Investigation Behavior. A further analysis examined the interaction of the users with the displayed system results. Especially, we looked at how often users zoomed in on a displayed keyframe in order to judge its relevance – zooming in on the keyframe and watching the video were both considered as zoom actions – and how often they selected shots as relevant for their search; this is different from shots being judged as *relevant* by the assessors. We mention here the results of the first test phase only and later discuss the differences to the second phase.

Our results indicate that both user groups perform almost the same number of zoom actions. Professional users investigate however shots much deeper in the ranked retrieved list, as the median rank of their zoomed and selected items is twice as high when compared to novice users. On average, the total number of zoom actions of users (irrespective of the user group they belong to) is twice as high as the number of select actions, which indicates that the initial result overview showing thumbnail keyframes is often insufficient to judge a shot on relevance.

Looking at the results of the second test phase, we observe the same difference between professional and novice users described above. The median rank of selected items for the archivists lies at 38, respectively 19 for novices. Compared to the first phase we further see a highly increased number of zoom actions due to the new mouse-over trigger in the search interface. In total the search logs of the second phase contained 4682 zoom actions, 317 select actions, but only 103 times users opened the detail view in order to watch the video shot. Hence, most shots were selected based on their keyframe rather than by watching the video. These numbers are, however, highly determined by the search interface. If the interface could also lower the burden to watch a shot as it was done for zooming the keyframes, we may expect more users to view a shot before selecting it as relevant.

6 Conclusions

In 2009, Wilkins et al. [16] presented an analysis of a multi-site video retrieval experiment using the TRECVID framework, where they discovered that non-expert

users generated very large performance fluctuations. We find a high variance in our experimental data as well, especially among the novice users. However, the primary purpose of our experiments is not a direct comparison between the performance of different video retrieval techniques, but rather identify differences in search behavior between the two user groups. While we cannot conclude that one technique performs consistently better than another one, we *can* draw conclusions such as “professional users apply concept-based searches significantly more often”, or, “novices are more likely to consider their information needs satisfied in spite of low recall”. Irrespective of the large variance between individual users in each group, clear patterns arise from observing their aggregated observations.

The evaluation of the different search methods used showed clearly that simple text search is employed as the default method, especially when starting a new search, even if other methods can be more effective, and, are also experienced as such. We expect that more training to gain a better knowledge of all available concepts, improved concept suggestion functionality, and the possibility to start a search by visual similarity initiated by an uploaded image (e.g., as the result of a web image search) are three directions that can help users make more effective use of the visual search techniques. One could argue that the observations in our data with respect to the effectiveness of visual search techniques are in line with the findings of recent evaluations of automatic systems at TRECVID. However, users in our experiments value the “query-by-concept” and “query-by-example” search methods more highly than those reported in earlier experiments.

The differences in search behavior of the two user groups in our study are most clearly visible when looking at the result investigation behavior. Professionals are willing to spend much more time to investigate the presented results, especially when this behavior leads to more and better results. A search system designed for professional archivists should therefore be optimized for high recall, and enable the user to easily investigate the relevance of found shots. For the VITALAS system, two straightforward interface improvements would be to let the user easily modify the size of shown keyframes, and to provide the possibility to play a shot by mouse-over (instead of the two clicks required now).

Although we tried to point out the importance of user studies conducted with real users – instead of the often presented results from experts being involved with the system development – it is at the same time undesirable for a realistic user study that professionals are not familiar with search system they have to use. While a short training session could overcome this problem partially, we anxiously await the time when we can repeat this study with professional archivists who have actually used visual search techniques in their daily work practice.

Acknowledgements. This paper was realised in part by funding from the European Union via the European Commission project Vitalas, contract no. 045389. We are grateful to our colleagues from INA, IRT and Beeld & Geluid, for arranging the tests involving archivists on their premises.

References

1. Boujemaa, N., Fauqueur, J., Ferecatu, M., Fleuret, F., Gouet, V., Le Saux, B., Sahbi, H.: Ikona: Interactive generic and specific image retrieval. In: Proceedings of the International Workshop on Multimedia Content-Based Indexing and Retrieval, MMCBIR 2001 (2001)
2. Christel, M.G.: Establishing the utility of non-text search for news video retrieval with real world users. In: Lienhart, R., Prasad, A.R., Hanjalic, A., Choi, S., Bailey, B.P., Sebe, N. (eds.) Proceedings of the 15th ACM International Conference on Multimedia, pp. 707–716. ACM, New York (2007)
3. Christel, M.G., Conescu, R.M.: Addressing the challenge of visual information access from digital image and video libraries. In: Marilino, M., Sumner, T., Shipman III, F.M. (eds.) Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2005), pp. 69–78. ACM, New York (2005)
4. Christel, M.G., Conescu, R.M.: Mining Novice User Activity with TRECVID Interactive Retrieval Tasks. In: Sundaram, H., Naphade, M.R., Smith, J.R., Rui, Y. (eds.) CIVR 2006. LNCS, vol. 4071, pp. 21–30. Springer, Heidelberg (2006)
5. Christel, M.G., Moraveji, N.: Finding the right shots: assessing usability and performance of a digital video library interface. In: Schulzrinne, et al. (eds.) [12], pp. 732–739
6. Christel, M.G., Yan, R.: Merging storyboard strategies and automatic retrieval for improving interactive video search. In: Sebe, N., Worring, M. (eds.) Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007), pp. 486–493. ACM, New York (2007)
7. Diou, C., Stephanopoulos, G., Dimitriou, N., Panagiotopoulos, P., Papachristou, C., Delopoulos, A., Rode, H., Tsirikika, T., de Vries, A.P., Schneider, D., Schwenninger, J., Viaud, M.-L., Saulnier, A., Altendorf, P., Schröter, B., Elser, M., Rego, A., Rodriguez, A., Martínez, C., Etxaniz, I., Dupont, G., Grilhères, B., Martin, N., Boujemaa, N., Joly, A., Enfciaud, R., Verroust, A., Selmi, S., Khadhraoui, M.: VITALAS at TRECVID-2009. In: Proceedings of the 7th TREC Video Retrieval Evaluation Workshop, TRECVID 2009 (2009)
8. Gauvain, J.-L., Lamel, L., Adda, G.: The LIMSI broadcast news transcription system. *Speech Communication* 37(1-2), 89–108 (2002)
9. Hauptmann, A.G., Christel, M.G.: Successful approaches in the TREC video retrieval evaluations. In: Schulzrinne, et al. (eds.) [12], pp. 668–675
10. Hiemstra, D., Rode, H., van Os, R., Flokstra, J.: PFTijah: text search in an XML database system. In: Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR 2006), pp. 12–17 (2006)
11. Hollink, L., Nguyen, G., Koelma, D., Schreiber, A., Worring, M.: Assessing user behaviour in news video retrieval. In: IEE Proceedings on Vision, Image and Signal Processing, pp. 911–918 (2005)
12. Schulzrinne, H., Dimitrova, N., Sasse, M.A., Moon, S.B., Lienhart, R. (eds.) Proceedings of the 12th ACM International Conference on Multimedia. ACM, New York (2004)
13. Snoek, C.G.M., van de Sande, K.E.A., de Rooij, O., Huurnink, B., Uijlings, J.R.R., van Liempt, M., Bugalho, M., Trancoso, I., Yan, F., Tahir, M.A., Mikolajczyk, K., Kittler, J., de Rijke, M., Geusebroek, J.-M., Gevers, T., Worring, M., Koelma, D.C., Smeulders, A.W.M.: The MediaMill TRECVID 2009 semantic video search engine. In: Proceedings of the 7th TREC Video Retrieval Evaluation Workshop, TRECVID 2009 (2009)

14. Snoek, C.G.M., van de Sande, K.E.A., de Rooij, O., Huurnink, B., van Gemert, J.C., Uijlings, J.R.R., He, J., Li, X., Everts, I., Nedovic, V., van Liempt, M., van Balen, R., Yan, F., Tahir, M.A., Mikolajczyk, K., Kittler, J., de Rijke, M., Geusebroek, J.-M., Gevers, T., Worring, M., Smeulders, A.W.M., Koelma, D.C.: The MediaMill TRECVID 2008 semantic video search engine. In: Proceedings of the 6th TREC Video Retrieval Evaluation Workshop (TRECVID 2008) (2008)
15. Westman, S., Lustila, A., Oittinen, P.: Search strategies in multimodal image retrieval. In: Lalmas, M., Tombros, A., Borlund, P., Schneider, J.W., Kelly, D., Feather, J., de Vries, A.P. (eds.) Proceedings of the 2nd International Conference on Information Interaction in Context (IiX 2008), pp. 13–20. ACM, New York (2008)
16. Wilkins, P., Troncy, R., Halvey, M., Byrne, D., Amin, A., Punitha, P., Smeaton, A.F., Villa, R.: User variance and its impact on video retrieval benchmarking. In: Marchand-Maillet, S., Kompatsiaris, Y. (eds.) Proceedings of the 8th International Conference on Content-based Image and Video Retrieval (CIVR 2009), pp. 1–8. ACM, New York (2009)

An Affect-Based Video Retrieval System with Open Vocabulary Querying

Ching Hau Chan¹ and Gareth J.F. Jones^{1,2}

¹ Centre for Digital Video Processing

² Centre for Next Generation Localisation,
School of Computing, Dublin City University, Dublin 9, Ireland
gjones@computing.dcu.ie

Abstract. Content-based video retrieval systems (CBVR) are creating new search and browse capabilities using metadata describing significant features of the data. An often overlooked aspect of human interpretation of multimedia data is the affective dimension. Incorporating affective information into multimedia metadata can potentially enable search using this alternative interpretation of multimedia content. Recent work has described methods to automatically assign affective labels to multimedia data using various approaches. However, the subjective and imprecise nature of affective labels makes it difficult to bridge the semantic gap between system-detected labels and user expression of information requirements in multimedia retrieval. We present a novel affect-based video retrieval system incorporating an open-vocabulary query stage based on WordNet enabling search using an unrestricted query vocabulary. The system performs automatic annotation of video data with labels of well defined affective terms. In retrieval annotated documents are ranked using the standard Okapi retrieval model based on open-vocabulary text queries. We present experimental results examining the behaviour of the system for retrieval of a collection of automatically annotated feature films of different genres. Our results indicate that affective annotation can potentially provide useful augmentation to more traditional objective content description in multimedia retrieval.

Keywords: affective computing, information retrieval, multimedia data, open vocabulary querying, automatic annotation.

1 Introduction

The amount of professional and personal multimedia data in digital archives is currently increasing dramatically. With such large volumes of data becoming available, manually searching for a multimedia item from within a collection, which is already a time-consuming and tedious task, is becoming entirely impractical. The solution to this problem is to provide effective automated or semi-automated multimedia retrieval and browsing applications for users. This of course requires the data to be annotated with meaningful features to support user search. Unfortunately, it is unrealistic to expect all multimedia data to be

richly annotated manually therefore automated content analysis tools are vital to support subsequent retrieval and browsing.

According to [1] and [2] annotation can be differentiated into 3 levels as follows: labels at the lowest level (feature level) are primitive features such as shot cuts and camera motion, the next level is logical features (cognitive level) involving some degree of logical inference describing the content such as “red car below a tree” and finally the highest level (affective level) contains the most abstract features that involve some degree of subjectivity, such as “calm scene” or “funny face”. Current multimedia retrieval systems are generally based on low-level feature-based similarity search. These systems are limited in terms of their interpretation of the content, but they are also difficult for non-expert users to work with since they typically want to retrieve information at the cognitive or affective level rather than working with low-level image features [3]. The difference between the low-level information extracted from multimedia data and the interpretation of the same data by the user in a given situation is identified as a *semantic gap* [4]. Developing methods to close the semantic gap to support more powerful and intuitive search of multimedia content is one of the ongoing research challenges in multimedia information retrieval. In complementary research, the field of *affective computing* focuses on the development of human-centered systems that can contribute to bridging this semantic gap [5]. For example, methods based on affective computing by allowing could enable users to query a system on a higher level of abstraction such as “find some exciting videos” instead of a low-level query describing features associated with the concept of “exciting” such as “rapid motion”, “shot cuts” and “elevated audio energy”.

In this paper we present work on a novel system designed to be a step towards providing this higher abstraction through an affect-based annotation of video content. The system automatically extracts a range of low-level audio and video features and then uses these to assign a set of affective verbal labels to the content. Video retrieval is then enabled using a system based on the Okapi retrieval model with an additional query pre-processing stage based on WordNet to provide open-vocabulary querying. Experimental retrieval results on a wide ranging collection of commercial movies show that affective annotation has the potential to augment existing multimedia search based on low-level objective descriptive features of the content.

This paper is organized as follows: Section 2 describes our affect extraction and labeling method for video data, Section 3 gives a summary description of the Okapi retrieval method used in our system, Section 4 presents our experimental investigations based on a movie collection, and Section 5 summarizes our conclusions and outlines possible directions for future work.

2 Affect Extraction and Labeling

One step towards bridging the semantic gap between user needs and detected low-level features is to combine these features to infer some form of higher-level features to which non-expert users can relate [6]. This method is favoured

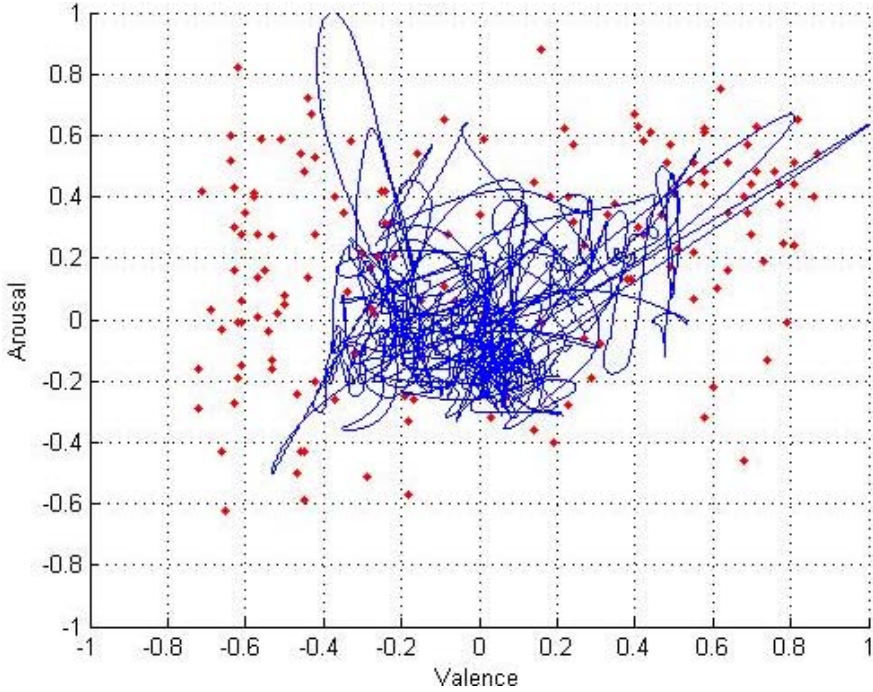


Fig. 1. Affect curve (line) mapped onto 151 emotional verbal labels (dots)

because it splits the semantic gap problem into two stages: mapping low-level features to immediate semantic concepts, and mapping these semantic concepts to user needs. As outlined in the previous section, existing research on multimedia content analysis and retrieval has concentrated largely on recognition of objective features describing what is observed in the content [7]. Such work includes scene segmentation [8][9], object detection [10], and sports highlight detection [11]. Our research aims to complement these objective features by using low-level features to describe an affective interpretation of the content. A detailed description of our annotation method is described in [12]. In this section we summarise our approach and in the following sections extend this existing work into an affect-based retrieval system.

The following subsections give a summary description of the annotation procedures for our affect-based retrieval system.

2.1 Modeling Valence and Arousal

The affective dimension of a video describes information of its emotive elements which are aimed at invoking certain emotional or affective states that humans can naturally identify. Therefore affective labels of the multimedia content relate to the affective states that the creator of the content is seeking to elicit in the viewers. Including such labels in multimedia indexes would enable human users

of multimedia retrieval systems to include expression of affective states as part of queries expressing their information needs. However, since affective states are subjective, even when presented with a set of affective labels individual users may select a different, but usually related, label to describe the same affective state that they associate with the desired multimedia content.

Research into human physical and cognitive aspects of emotion can help us to model affective features. One such model which is extremely useful in this context, is the Valence-Arousal-Dominance (VAD) emotion representation described by Russell and Mehrabian [13] which breaks emotions into 3 independent components. In the VAD model the 3 independent and bipolar dimensions represent all emotions experienced by humans. The three components are defined as follows:

- Valence: Measures the level of pleasure - displeasure being experienced. This ranges from a “positive” response associated with extreme happiness or ecstasy through to a “negative” response resulting from extreme pain or unhappiness.
- Arousal: This is a continuous measure of alertness ranging from one extreme of sleep through to intermediate states of drowsiness and alertness and finally frenzied excitement at the other end of the scale.
- Dominance: Dominance (or control) is a measure of a person’s perceived control in a situation. It can range from feelings of a total lack of control or submissiveness to the other extreme of complete control or influence over their situation or environment. It has been observed that this dimension plays a limited role in affective analysis of video, and we follow previous work in concentrating only on valence and arousal [2].

In order to extract the affective content contained in video data, we first perform low-level feature extraction. Extended features are then combined to describe valence and arousal levels of the data as follows:

- Valence is modeled as a weighted sum of colour brightness and colour saturation from the visual stream and pitch from the audio stream with each weighting following findings reported in [14].
- Arousal is modeled as an equally weighted sum of global motion and shot cut rate from the visual stream, and energy from the audio stream as described in [2].

Each of the low-level features are subjected to a smoothing function and normalized in the range -1 and +1 to fulfill [2]’s comparability, compatibility, and smoothness criterion. The arousal and valence stream outputs of this process can be illustrated on a 2D VA plot showing an affect curve which plots valence and arousal against each other as illustrated in Figure 1. The affect curve illustrates the evolution of affective states contained in the data stream over time.

2.2 Verbal Labeling of the Affect Curve

To automatically annotate videos with affective labels, the affect curve is populated with emotional verbal labels from the findings of [13]. In this study averaged

arousal, valence, and dominance values were assigned to 151 verbal labels by a group of human assessors. Figure 1 shows the 151 emotional verbal labels plotted as individual dots on the affect curve. The values are normalized in the range -1 to +1 so that the verbal labels can be mapped to the affect curve. It can be seen from Figure 1 that the labels are quite evenly distributed in the VA space enabling us to describe a wide range of emotions. Each point on the affect curve can be associated with the spatially closest label.

Automatically detecting the low-level visual and audio features of video data and processing it to obtain its affect curve allows us to annotate each frame of the video with a affective label. Taken over the duration of a multimedia document this generates a sequence of affect labels for the content. Therefore a simple frequency count of re-occurring labels can describe the major affective state(s) or emotional content of each part of the multimedia data. 151 verbal labels gives a quite fine level of labeling granularity, but since affective interpretation is subjective, we can also use the label stream to refer to alternative labels by choosing second or third closest labels. To study the granularity of labeling and the overall quality of affective labeling, we explored an additional two sets of verbal labels with coarser granularity. These consisted of 22 labels suggested by [15] and 6 labels based on word described in [16]. A similar approach was used in [17] using manual placement of 40 labels for the FEELTRACE system. A comparative investigation of these different labeling schemes is described in [12], as might be anticipated the overall conclusion was that there is a trade off between granularity of labels and reliability of individual labels. A larger number of labels mean a greater degree of expressivity, however inevitably the accuracy of label assignment will be reduced. Our experiments described later illustrate the need for a larger annotation vocabulary to support effective open-vocabulary search.

3 Information Retrieval and Document Matching

The classic information retrieval (IR) problem is to locate desired or relevant documents in response to a user information need expressed using a search query consisting of a number of words or search terms (which may be stemmed or otherwise pre-processed). Matching the search terms from the query with the terms within the documents then retrieves potentially relevant documents. Documents are ranked according to a query-document matching score measuring potential likelihood of document relevance. The user can then browse the retrieved documents in an attempt to satisfy their information need.

Using this principle with each visual and audio frame of the multimedia data with an affective label, the multimedia data can be thought of as a document containing re-occurring words where the frequency is directly related to the degree to which the affect associated with the label is present in the multimedia item. This enables us to perform experiments into the retrieval of multimedia data from the perspective of affective content using a text IR model. Thus our system is based on entry of a text query which is used with an IR model to match the query with the system-detected affect labels to retrieve a ranked list of potentially relevant videos.

3.1 Okapi BM25 Information Retrieval Model

A number of IR algorithms have been developed which combine various factors to improve retrieval effectiveness. The effectiveness of these methods has been evaluated extensively using text test collections such as those introduced at the TREC evaluation workshops, see for example [18]. One of the most consistently effective methods for text retrieval is the Okapi BM25 IR model [19]. BM25 is a classical term weighting function. For a term i in a document j , the BM25 combined weight $CW(i, j)$ is:

$$CW(i, j) = \frac{CFW(i) \times TF(i, j) \times (K1 + 1)}{K1 \times ((1 - b) + (b \times (NDL(j)))) + TF(i, j)}$$

where $K1$ and b are tuning constants. $CFW(i) = \log(N/n(i))$ is the collection frequency weight where N is the total number of documents in the collection and $n(i)$ is the total number of documents containing term i . $TF(i, j)$ is the frequency of term i in document j . The BM25 formula overall ensures that the effect of term frequency is not too strong, and for a term occurring once in a document of average length that the weight reduces to a function of $CFW(i)$ for a document of average length. The overall matching score for a document j is simply the sum of the weights of the query terms present in the document. Documents are ranked in descending order of their matching score, for presentation to the user. The tuning constant $K1$ modifies the extent of the influence of term frequency. The constant b , which ranges between 0 and 1, modifies the effect of document length. If $b = 1$ the assumption is that the documents are long simply because they are repetitive, while if $b = 0$ the assumption is that they are long because they are multi-topic. Thus setting b towards 1, such as $b = 0.75$ will reduce the effect of term frequency on the grounds that it is primarily attributable to verbosity. If $b = 0$, there is no length adjustment effect, so greater length counts for more, on the assumption that it is not predominantly attributable to verbosity.

Since Okapi BM25 has been shown to be effective in many retrieval settings, we adopt it in our affect-based retrieval system.

3.2 Open-Vocabulary Query

In standard text IR, documents and queries both use an open vocabulary. For a well constructed IR system, the success of an IR system relies on there being a good match between words appearing in relevant documents and the submitted search request.

The affect labeling described in section 2.2 is limited to 151 labels. While the 151 labels cover a wide range of possible affective states, it is likely that users will often use query words that are not part of this list. This mismatch between user query and detected affect labels in the system will mean that the relevant documents may often not be retrieved, or be retrieved unreliably at a reduced rank. In order to address this problem we use a novel solution to enable open-vocabulary querying. In this method a measure of relatedness is

calculated between each query word entered by the user and the list of affective labels used by the system. This ensures that even if the query word is not one of the annotated multimedia labels, the system is able to produce a ranked list of closest match multimedia documents for the user.

We use WordNet [20], a freely available lexical database/dictionary consisting of nouns, verbs, adjectives, and adverbs organized into a network of related concepts called synonym sets to provide our open-vocabulary word relatedness scoring. This provides us with a tool to measure semantic relatedness or similarity between different words. Our system uses WordNet::Similarity, a freely available Perl module that implements the similarity and relatedness measures in WordNet [20].

A measure of similarity quantifies how much two concepts (or words) are alike based on the information contained in WordNet’s relations or hierarchy. Due to the organization of words into synonym sets, two words can be said to be similar if counting the distance of the relation or hierarchy results in a small distance. Extending this further, a similarity measure can be derived by counting the path lengths from one word to another, utilizing the *is-a* relation. [21] presents an algorithm to find the shortest path between two concepts and scales this value by the maximum path length D in the *is-a* hierarchy in which they occur. A different take on a similarity measure is proposed by [22], which uses knowledge of a corpus to derive a similarity measure. Their similarity measure is guided by the intuition that similarity between a pair of words may be judged by the extent to which they share information.

However *is-a* relations in WordNet do not cross part-of-speech boundaries, so such measures are limited to judging relationships between noun pairs and verb pairs. The system presented in this paper relies on adjective words that describe emotions such as “happy”, “joyful” and “sad” in order to describe affective states. Adjectives and adverbs in WordNet are not organized into *is-a* hierarchies, but can still be related through antonyms and *part-of* relations, called measures of relatedness. For example, a “wheel” has a *part-of* relationship with a “car”, and “happy” is the opposite of “unhappy”.

We use the measure of relatedness proposed by [23], where the idea of semantic relatedness is that two words are semantically close if their WordNet synonym sets are connected by a path that is not too long and does not change direction too often. For every query label, we use WordNet::Similarity to compare it to the 151 affective labels. If a label is highly related it will score higher, while query words that are found in the list of 151 affect labels are given the maximum score.

3.3 Query Processing Strategies

We explored six different query processing strategies in a known-item search task discussed in the next section. The first four strategies use the open-vocabulary query processing to map the user query words to the annotation system’s 151 affective labels to form a final query that is fed into the IR system for retrieval. The four strategies are referred to as: “Unweighted full expansion”, “Weighted

full expansion”, “Unweighted best expansion”, and “Weighted best expansion”. The remaining two strategies are referred to as: “Reweighted” and “Bypass”.

- Full expansion means that all labels that have relatedness scores above 0 for each user-supplied query word were fed into the IR model. This in effect fully expands the user query to the all available labels, hence “full expansion”.
- Best expansion means that only affect labels with the highest (best) relatedness score for each user query word were input to the IR model. Sometimes a query label will have 2 labels with identical highest relatedness scores, in these cases both labels were used.
- Unweighted or Weighted determines whether the final expanded query input to the IR model was weighted according to the relatedness scores calculated from WordNet. For the Unweighted strategy, every word included in the query was said to be equally important. For the Weighted strategy, each query word’s relatedness score was multiplied by the combined weight $CW(i, j)$ of the IR model to generate the ranked retrieval list.

The following equations show the mathematical formulae for the modified BM25 weights for the full expansion strategies, if the relatedness score $rel_{HS}(i_R, i) \neq 0$, where i_R is the relevant term.

- Unweighted full expansion:

$$CW_{UW}(i, j) = \sum_{i_R} CW(i_R, j)$$

- Weighted full expansion:

$$CW_W(i, j) = \sum_{i_R} rel_{HS}(i_R, j) \times CW(i_R, j)$$

The following strategies are applied for the best expansion strategies if the relatedness score $rel_{HS}(i_R, i) = \max$, where i_R is the scored as the most related term.

- Unweighted best expansion:

$$CW_{BUW}(i, j) = \sum_{\substack{i_R \\ rel_{HS}(i_R, i) = \max}} CW(i_R, j)$$

- Weighted best expansion:

$$CW_{BW}(i, j) = \sum_{\substack{i_R \\ rel_{HS}(i_R, i) = \max}} rel_{HS}(i_R, j) \times CW(i_R, j)$$

The fifth strategy is a modification of the “Weighted best expansion” strategy where the relatedness score was first raised by an exponent value X before it was multiplied with the combined weights of the IR model. This gives the higher range of the relatedness scores heavier weights and emphasis, if the relatedness score $rel_{HS}(i_R, j) = \max$, where i_R is the relevant term and X is the exponent value.

- Reweighted best expansion:

$$CW_{BRW}(i, j) = \sum_{i_R} rel_{HS}(i_R, j)^X \times CW(i_R, j)$$

The sixth strategy called “Bypass” is to bypass the open vocabulary query mapping of the system and directly feed the queries into the IR model to generate a ranked list, therefore any query words not found in the system’s list of affect labels are simply ignored. The following combined weight is obtained if query term i is found in the list of affective labels, where i_R is the relevant term.

- Bypass:

$$CW_B(i, j) = \sum_{i_R} CW(i_R, j)$$

In each case the standard $CW(i_R, j)$ weight in the BM25 function is replaced by the relevant modified version.

4 Experimental Investigation

The experimental investigation presented here explores the behaviour and potential of our affect-based retrieval system for searching a collection consisting of a variety of commercial Hollywood movies. Movies of this type represent a compelling source of video data for an affect-based system due to the richness of their emotional content. They are additionally suitable for our initial study of affect-based search since emotional content is much more pronounced in movies than in other video material, making it easier to determine what emotions a movie is trying to project. A total of 39 movies were processed covering a wide range of genres from action movies to comedy and horror movies. This amounted to approximately 80 hours of data comparable to video evaluation campaigns such as TRECVID [\[24\]](#). Each movie was broken up into 5 minute clips, giving a total of 939 film clips.

4.1 Known-Item Search Task

A known-item search task was performed to measure the system’s effectiveness at retrieving and locating the original film clip described by a text query, from the database of films using a textual description of that particular clip’s emotional content labelled using our affect label assignment system.

In order to generate the test set, 8 volunteers were each randomly assigned a unique set of 5 film clips. They viewed each of these clips and then created an open-vocabulary affective textual description of it. These descriptions were collected together as a set of 40 search queries for the known-item search task. Depending on the query processing strategy used, the user query was

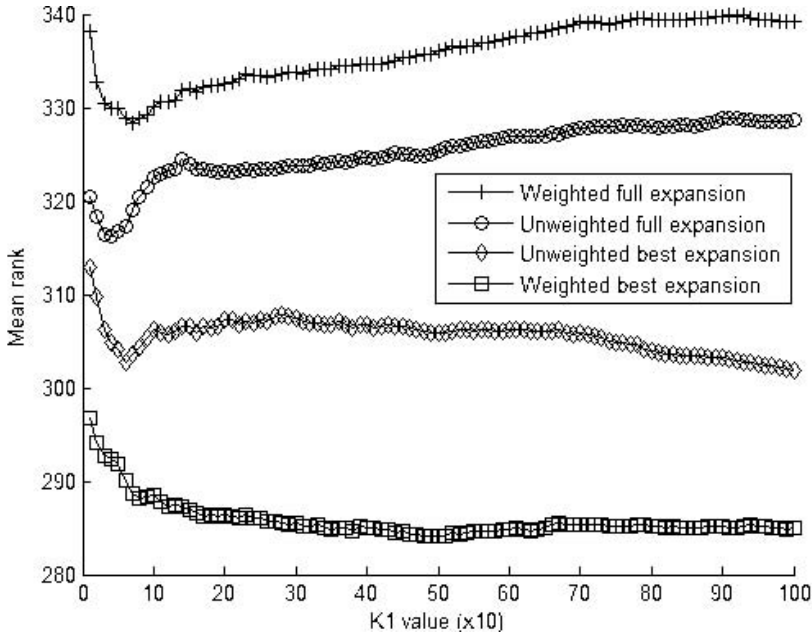


Fig. 2. Mean rankings for the 4 query processing strategies

mapped to the system’s affective list of 151 labels, and these labels used as the final query. The system generated a ranked list of clips from the database of movies using the Okapi BM25 IR model. The ranked list sorted the clips in order of relevance. From this ranked list, the original clip’s position on this list was identified. The higher its position in the list, the better the retrieval performance.

4.2 Experimental Results

Figure 2 shows how different $K1$ values affect the mean rankings of the relevant film clip for the first 4 query processing strategies. A thousand runs were performed using the system to automatically calculate the mean rank for the 40 queries. The b value was set to 0 in all cases. There is a noticeable dip in mean rank for all strategies when the $K1$ value reaches 51. It can be observed that the best mean rank achieved by the “Weighted best expansion” strategy is when $K1$ is at the value 474.

Figure 3 shows that the best mean ranks of relevant items was achieved when the b value was set to 0, except for the “Unweighted best expansion” strategy where the mean rank of the relevant clips were degraded when b was set to 0. When the b values changes from 0 to 1, the mean ranks does not appear to change. Closer inspection of the values reveal that the mean ranks do change with different b values, but that the variation was too small to be noticeable on the graph. The b value in the BM25 model relates to the topical structure of

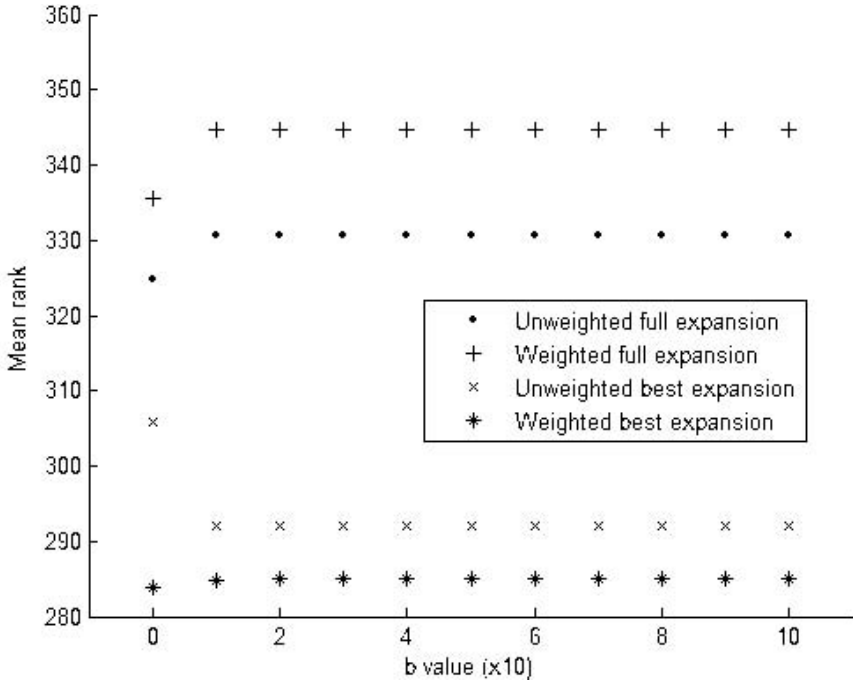


Fig. 3. Mean rankings for the 4 query processing strategies

Table 1. Number of returned results for the query processing strategies

Total queries: 40	Unweighted full exp.	Weighted full exp.	Unweighted best exp.	Weighted best exp.	Bypass
Number of returned results	32	32	26	26	3
Recall	0.8	0.8	0.65	0.65	0.08

documents and document length, since all documents were of the same length with similar topical structuring, it is unsurprising that b is not a significant component in retrieval effectiveness for this task.

Table 1 shows the number of queries for which the system successfully retrieved the target clip using the 5 query processing strategies. The two “full expansion” strategies retrieved 32 clips out of 40, giving the best recall rate of 0.8. The “Bypass” strategy only retrieved 3 clips, with the lowest recall rate of 0.08. This indicates that as the user word query is mapped to an increasingly smaller number of labels to form the final query, the recall rate drops. Note that the failure to retrieve a clip at any rank indicates that none of the query words were contained in the label from the affective label list automatically assigned to the target clip in the analysis stage.

Table 2. Comparison of the ranks for the query processing strategies

Queries relevant result for “Bypass”	Unweighted full exp.	Weighted full exp.	Unweighted best exp.	Weighted best exp.	Bypass
Query 4	346	451	346	451	58
Query 14	301	211	182	94	6
Query 36	51	13	12	14	7

Table 3. Mean rank and mean-reciprocal-rank (MRR) for the query different expansion strategies for 26 queries retrieving relevant items with Best Expansion methods

	Unweighted full exp.	Weighted full exp.	Unweighted best exp.	Weighted best exp.	Reweightd best exp. (16)
Mean rank	301.9	284.8	307.8	308.5	271.0
MRR	0.027	0.019	0.022	0.0221	0.025

Table 2 shows the individual retrieved rank results for the three queries for which the relevant item was retrieved by the “Bypass” strategy and their ranks across the different strategies. It can be observed that the “Bypass” strategy achieved the best ranks for all three of these queries. These results illustrate that exact matches with the affect label list in the query can be very effective for retrieval. The difficulty with limiting the query vocabulary to only these labels is that users are likely to find this list of words constraining and difficult to use in describing their information needs. Hence expansion to alternative labels is needed for good recall levels, but at the cost of greatly degraded average rank at which relevant items are retrieved.

Mean-reciprocal-rank (MRR) is calculated as the mean of the reciprocal of the rank of retrieved relevant items, using 0 for queries for which the known-item is not retrieved. Compared to the mean rank, the MRR has the effect of not punishing a system excessively for retrieval of individual items at very low rank. Thus, it gives a better indication of the average performance across a query set. The nearer the MRR is to 1.0, the better the system is performing on average. Table 3 shows the mean rank and MRR results for the 4 expansion strategies for the 26 topics for which the “best expansion” strategy retrieves the relevant item, these are a subset of the 32 queries for which the relevant item is retrieved by “full expansion”. This subset was used to enable a direct comparison of results for the different strategies. It can be seen that the best MRR is given by “Unweighted full expansion”. Examination of individual results shows that with “Unweighted full expansion” for a small number of queries the relevant item is retrieved at a much higher rank than with the other strategies, although on average it performs less well than the other strategies, leading to its better MRR and worse mean rank effectiveness than the “Best” expansion strategies.

Figure 4 shows the mean ranks for the re-weighted (fifth) “best expansion” strategy, where the numerical value enclosed in round brackets is the exponent value. It can be seen that as the value of the X parameter increases, the mean

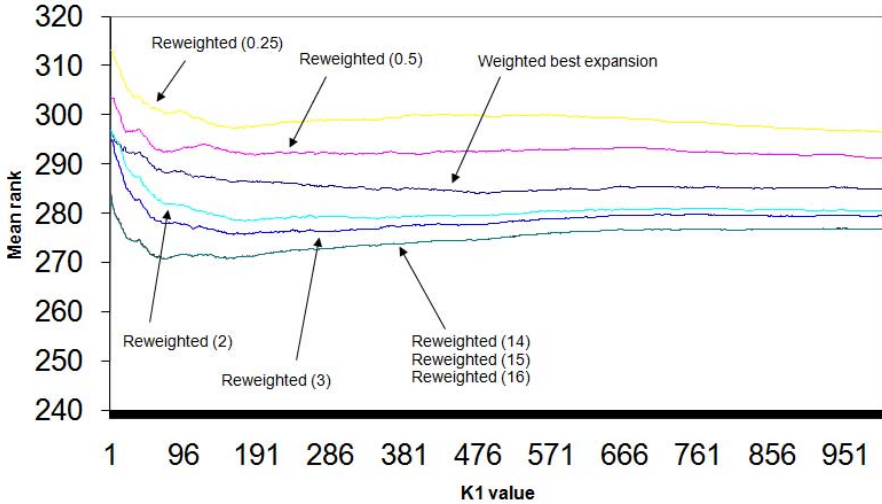


Fig. 4. Mean rankings for different exponent values for the query processing strategies

ranks are improved over the original strategy. The mean ranks improvement stops when the exponent value reaches $X = 14$, as can be seen in the figure, the lines ($X = 15$ and $X = 16$) are almost identical to the line for $X = 14$. In addition, the best $K1$ value for the re-weighted strategies was found to be 71. The final column of Table 3 shows the mean rank values and MRR for the re-weighted “best expansion” for $k = 71$. It can be seen that the exponential function gives improvement in both the mean ranks and MRR values. This is the best mean rank result, while the MRR result is still slightly less than that achieved with “unweighted full expansion” for the reasons given earlier.

5 Conclusions and Further Work

The results of experiments reported in this paper show that video clips which had been described by users using text queries can be retrieved with a measure of consistency for affect-based user queries. While the WordNet expansion is shown to improve recall, it is clear that where the affect label vocabulary covers the query words that retrieval effectiveness is better. Thus, it would appear that rather than seeking an improvement through increased technical sophistication, the most effective strategy would be to generally increase the affective label set that can be placed on the VA plot. While in the past such an endeavour would have been costly and difficult to arrange, crowdsourcing methods such as Mechanical Turk¹ could potentially make it relatively straightforward to gather valence and arousal values for very many words averaged across a large number of people. A more sophisticated method of assigning labels to video data would

¹ <https://www.mturk.com>

then be required since the single assignment of a label based on VA proximity will not be sufficiently accurate, labels might be clustered to an average location or perhaps multiple labels might be assigned based on some proximity measure.

Also since affect is a subjective interpretation of an important, but limited, dimension in describing multimedia content, we believe that affect-based annotation is more likely to be used most effectively to augment existing objective multimedia content retrieval systems, rather than to be used independently. Further work is planned to explore how this alternative dimension of indexing and search can be incorporated into existing multimedia retrieval systems.

Acknowledgements. This research is partially supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) at Dublin City University.

References

1. Eakins, J.P.: Automatic image content retrieval - are we getting anywhere? In: Proceedings of the 3rd International Conference on Electronic Library and Visual Information Research, pp. 123–135. De Montfort University, Milton Keynes (1996)
2. Hanjalic, A., Qun Xu, L.: Affective Video Content Representation and Modeling. *IEEE Transactions on Multimedia* 7(1), 143–154 (2005)
3. Lee, H., Smeaton, A.F., McCann, P., Murphy, N., O'Connor, N., Marlow, S.: Físchlár on a PDA: A Handheld User Interface to a Video Indexing, Browsing, and Playback System. In: ERCIM Workshop User Interfaces for All, Florence, Italy (2000)
4. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(12), 1349–1380 (2000)
5. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based Multimedia Information Retrieval: State of the Art and Challenges. *ACM Transactions on Multimedia Computing, Communications and Applications* 2(1), 1–19 (2006)
6. Hauptmann, A.G.: Lessons for the future from a decade of informedia video analysis research. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 1–10. Springer, Heidelberg (2005)
7. Hauptmann, A.G., Christel, M.G.: Successful Approaches in the TREC Video Retrieval Evaluations. In: Proceedings of the Twelfth ACM International Conference on Multimedia 2004, New York, NY, USA, pp. 668–675 (2004)
8. Zhang, T., Jay Kuo, C.-C.: Content-Based Audio Classification and Retrieval for Audiovisual Data Parsing. Kluwer Academic Publishers, Dordrecht (2001)
9. Zhai, Y., Shah, M., Rasheed, Z.: A Framework for Semantic Classification of Scenes using Finite State Machines. In: Proceedings of the Conference for Image and Video Retrieval, Dublin, Ireland, pp. 279–288 (2004)
10. Browne, P., Smeaton, A.F.: Video Information Retrieval Using Objects and Ostensive Relevance Feedback. In: ACM Symposium on Applied Computing, Nicosia, Cyprus, pp. 1084–1090 (2004)
11. Sadlier, D., O'Connor, N.: Event Detection based on Generic Characteristics of Field Sports. In: IEEE International Conference on Multimedia and Expo. (ICME 2005), Amsterdam, The Netherlands, pp. 759–762 (2005)

12. Jones, G.J.F., Chan, C.H.: Affect-Based Indexing for Multimedia Data. In: Maybury, M.T. (ed.) *Multimedia Information Extraction*. IEEE Computer Society Press, Los Alamitos (2011)
13. Russell, J.A., Mehrabian, A.: Evidence for a Three-Factor Theory of Emotions. *Journal of Research in Personality* 11, 273–294 (1977)
14. de Kok, I.: A Model for Valence Using a Color Component in Affective Video Content Analysis. In: *Proceedings of the Fourth Twente Student Conference on IT, Faculty of Electrical Engineering, Mathematics and Computer Science*. University of Twente, The Netherlands (2006)
15. Salway, A., Graham, M.: Extracting Information about Emotions in Films. In: *Proceedings of the Eleventh ACM International Conference on Multimedia 2003*, Berkeley, CA, USA, pp. 299–302 (2003)
16. Ekman, P., Friesen, W.V.: *Facial Action Coding System*. Consulting Psychologists Press Inc., Palo Alto (1978)
17. Cowie, R., Douglas-Cowie, E., Savvidou, S., McMahon, E., Sawey, M., Schröder, M.: ‘FEELTRACE’: An Instrument for Recording Perceived Emotion in Real Time. In: *Proceedings of the ISCA Workshop on Speech and Emotion: A Conceptual Framework for Research*, Belfast, U.K, pp. 19–24 (2000)
18. Harman, D. K.: *The Fifth Text Retrieval Conference (TREC-5)*. National Institute of Standards and Technology, Gaithersburg(1997)
19. Robertson, S.E., Spärck Jones, K.: Simple, proven approaches to text retrieval, Technical Report, TR356, Cambridge University Computer Laboratory (1997)
20. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:Similarity - Measuring the Relatedness of Concepts. In: *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, San Jose, CA, USA (2004)
21. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: *WordNet: An Electronic Lexical Database*, pp. 265–283. MIT Press, Cambridge (1998)
22. Resnik, P.: Using information content to evaluate semantic similarity. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 488–453 (1995)
23. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In: *Wordnet: An Electronic Lexical Database*, pp. 305–332. MIT Press, Cambridge (1998)
24. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and TRECVID. In: *MIR 2006: Proceedings of the Eighth ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, CA, USA, pp. 321–330 (2006)

A Comparison of Human, Automatic and Collaborative Music Genre Classification and User Centric Evaluation of Genre Classification Systems

Klaus Seyerlehner¹, Gerhard Widmer^{1,2}, and Peter Knees¹


¹ Dept. of Computational Perception, Johannes Kepler University, Linz, Austria
<http://www.cp.jku.at>

² Austrian Research Institute for AI, Vienna, Austria
<http://www.ofai.at>

Abstract. In this paper two sets of evaluation experiments are conducted. First, we compare state-of-the-art automatic music genre classification algorithms to human performance on the same dataset, via a listening experiment. This will show that the improvements of content-based systems over the last years have reduced the gap between automatic and human classification performance, but could not yet close this gap. As an important extension to previous work in this context, we will also compare the automatic and human classification performance to a collaborative approach. Second, we propose two evaluation metrics, called *user scores*, that are based on the votes of the participants of the listening experiment. This user centric evaluation approach allows to get rid of predefined ground truth annotations and allows to account for the ambiguous human perception of musical genre. To take genre ambiguities into account is an important advantage with respect to the evaluation of content-based systems, especially since the dataset compiled in this work (both the audio files and collected votes) are publicly available.

Keywords: genre classification, user centric evaluation.

1 Introduction

Although genre definitions and annotations are somewhat subjective, genre categorizations or genre hierarchies are often used to organize large scaled music collections, as there seems to be some general consensus on genre annotations, at least to a certain degree. In music information retrieval (MIR), genre labels often serve as ground truth information, most notably to evaluate automatic genre classification systems, music similarity algorithms and music recommender systems. While publicly available genre classification datasets and also the annual *Music Information Retrieval Evaluation eXchange* (MIREX)  make the numerous proposed systems more comparable to each other in terms of quality, there

¹ <http://www.music-ir.org/mirexwiki>

exists little work on making the evaluated systems comparable to human performance on the same task. To improve the comparability of automatic and human classification accuracy, we have conducted a listening experiment. This allows to compare the classification results of human listeners to those of state-of-the-art automatic genre classification algorithms. Furthermore, we will show that the collaborative result of the participants outperforms both automatic methods and individual human performance. While the collaborative result can be regarded as an upper bound on the achievable classification accuracy on this dataset, it also shows that collaborative techniques clearly outperform content-based approaches. Furthermore, the dataset containing both the full length tracks and the genre votes by the participants is publicly available from the first author's personal webpage. This will be useful to improve the evaluation of genre classification algorithms, because on the basis of such data one can define user centric evaluation metrics - so called *user scores*. The main advantage of user centric evaluation metrics is that one can account for genre ambiguities derived from the user votes whenever two automatic systems are compared.

The rest of the paper is organized as follows. First, in section 2 we report on the conducted listening experiment and point out the difference to the only related work by Lippens et al. [10]. In section 3 we then present the results obtained by the individual participants, briefly introduce five automatic classification methods and two collaborative approaches and compare the performance of these approaches to the performance of the individual participants. In section 4 we then discuss how the collected genre information can also be used to define two user centric evaluation metrics and present results for the automatic classification methods using the proposed evaluation criteria. Finally, we conclude on the obtained results in section 5.

2 The Listening Experiment

In general genre as an evaluation criterion is a well-discussed topic [6] [4] [18] [3] [11] and it is broadly accepted in Music Information Retrieval (MIR) as an evaluation criterion for content-based systems. Thus, there exist numerous publications focuses on comparing automatic systems to each other using genre information. There also exists some scientific work on evaluating the human abilities to classify music into genres. Most notably Gjerdingen et al. in [7] showed that humans are very fast at classify music into genres. About 300ms of audio are enough for humans to come up with the same categorization decision as with 3000ms of audio. Bella et al. in [2] investigated the human ability to classify classical music into sub-genres. Furthermore, Gaus et al. [8] study the effect of rhythm and timbre modifications on the human music genre categorization ability. They find that timbre feature provide more genre discrimination power than rhythm.

However, there exists little work on **comparing** automatic to human performance on the same genre classification task. In [17] Soltau et al. mentioned that the genre confusions of a conducted listening experiment are similar to those of a proposed automatic system, but no evaluation to directly compare human to

automatic performance was conducted. The only work that really focuses on a comparison of human to automatic classification performance we are aware of is the work of Lippens et al. [10] and dates back to 2004. In [10], a listening experiment is conducted where 27 human listeners manually classified a collection of 160 songs (the “MAMI dataset”), into 6 possible genres by listening to 30 seconds excerpts. The average performance of the participants (76%) is then compared to an automatic classification approach with a classification performance of 57%, and the baseline accuracy (26%). Unfortunately, the MAMI dataset and the survey data are not publicly available. To be able to also compare state-of-the-art systems to human classification, we decided to rerun a listening experiment quite similar to the one presented in [10]. In this listening experiment 24 persons were asked to do exactly the same task the machine was asked to solve, namely to categorize a set of songs into 19 genres. The participants of this survey were aged in between 20-40 and most of them had no specific musical background, but can be characterized as typical mainstream music consumers. The songs were drawn randomly from the “1517-Artists” dataset [15] in such a way that each genre is represented by 10 songs. The “1517-Artists” dataset itself consists of freely available songs from [download.com](http://music.download.com)²³ containing songs of both well-known and completely unknown artists. The genre labels were assigned by the artists of the songs. The genres and the number of tracks per genre of the subset used in the listening experiment are summarized in table 1. While it seems that just selecting 10 songs per genre is at the lower bound for a descriptive subset of a genre, the number of songs that can be used in such a listening experiment is of course limited by the available human resources. In our case many of the participants of the listening experiments reported that it took them many hours to complete the survey and far longer as expected.

Comparing the conducted listening experiment presented in this paper to the listening experiment in [10] there are some important differences in the data, the design of the experiment, and the analysis of the results:

- **Unique Artists**

To prevent artist effects and album effects [5], no two songs by one and the same artists are in the dataset used for the listening experiment. This is very important as artist and album effects can have a huge biasing influence on the obtained classification accuracies, especially on small datasets.

- **Number of Genres**

The number of genres (19) in our listening experiment is significantly larger, and the musical scope is broader than in the MAMI dataset.

- **Equal number of tracks per genres**

Each genre is represented by 10 representative songs, making this a balanced classification task that is not biased towards a popular, dominating genre like e.g. “Pop&Rock”.

² <http://music.download.com/>

³ The <http://music.download.com/> began redirecting all artist pages and category doors to corresponding pages on their sister music site Last.fm on March 2009.

– **Explicit Genre Annotations**

There exists a ground-truth genre label per songs that has been assigned by the artists that produced the songs via the music platform. The genre categories are the same as used by the music platform⁴.

– **Publicly Available Data**

The music files used in the presented experiment and the genre votes obtained through the listening experiment are both publicly available⁵. This will allow others to compare other methods not presented here to human performance in the future.

– **Collaborative Result**

In section 3.3 the votes of the subjects are used to collaboratively estimate a song’s genre. Thus, we are able to also compare the collaborative result of all subjects to both individual results as well as automatic classification systems.

It is important to note that we do not claim that the genre annotations of this dataset are particularly correct or that the genre taxonomy is perfectly consistent. In contrast we believe that genre and genre taxonomies by definition are ambiguous and inconsistent and good genre taxonomies need a careful design and should account for genre similarities [12]. However, it is important to see that for comparative evaluations like we perform in this paper annotation errors are not crucial as all evaluated approaches have to deal with the same annotation errors. With respect to genre inconsistencies we propose in section 4 to use so-called *user scores* as evaluation criteria, which allow to account for existing genre ambiguities.

The experiment was carried out as follows: Each participant was instructed to move the 190 anonymized full-length audio files into a set of folders representing the 19 genres, plus an extra folder “*other*” in case they had no idea what genre a song might belong to. Then a list of the files in the directory structure representing the genres was generated by a script and returned by each subject via e-mail. Finally, these files were parsed to obtain the votes of each individual.

3 Human, Automatic and Collaborative Classification

3.1 Human Classification

The collected information from the listening experiment is represented as a set T of tuples $t = (u_t, s_t, \hat{g}_t, g_t)$, where u_t (1 to 24) identifies the participant and s_t (1 to 190) the rated song. The ground truth genre of the song s_t is denoted $g_t \in G$, where G is the set containing the 19 ground truth genres. $\hat{g}_t \in G^+$ represents the genre predicted by participant u_t . G^+ is the set of genres plus the “*other*” category. The classification accuracy of subject u with respect to the given ground truth annotation is then given by

⁴ music.download.com

⁵ www.seyerlehner.info

Table 1. Genre distribution of the songs used in the listening experiment

Genre	#tracks
Blues	10
Country	10
Hip-Hop	10
Jazz	10
New Age	10
Reggae	10
Classical	10
Folk	10
Latin	10
Rock & Pop	10
Alternative & Punk	10
Electronic & Dance	10
R&B & Soul	10
World	10
Vocals	10
Children’s	10
Easy Listening	10
Comedy & Spoken Word	10
Soundtracks & More	10
total	190

$$acc_u = \frac{\sum_t^{\{t \in T | u_t = u\}} \hat{g}_t == g_t}{|\{t \in T | u_t = u\}|} \quad (1)$$

A look at Figure 1 shows that there is a huge variation in the performance of individual participants. Obviously the individual results heavily depend on the musical knowledge of the individuals. The worst participant exhibits a classification accuracy of 26%, which is still far better than the baseline (guessing), which would be 5%. The classification rate of the best individual is 71%. The average classification accuracy obtained by the participants is 55%, the median is also 55%. Figure 1 visualizes the classification accuracies achieved by the individual participants sorted from the worst to the best participant.

Aggregating the individual results of all users yields the overall classification result. Figure 2 shows the confusion matrix with respect to the ground truth. Altogether 55% of all song-genre assignments of the participants were correct. However the performance depends on the genre. While some genres seem to be well-defined (e.g. “Comedy&Spoken Word”, “Electronic&Dance”, “Hip-Hop”), there is almost no agreement among the participants for the genres “Folk” and “Vocals”. For the other genres the participants agree to a certain extent. The most significant genre confusions are “Folk” - “Vocals”, “Alternative&Punk” - “Rock&Pop”, “EasyListening” - “NewAge”, “Country” - “Folk”, “Blues” - “Jazz”, “Reggae” - “Hip-Hop” and “Latin” - “EasyListening” and vice versa. These confusions indicate genre ambiguities, but can also be interpreted as some

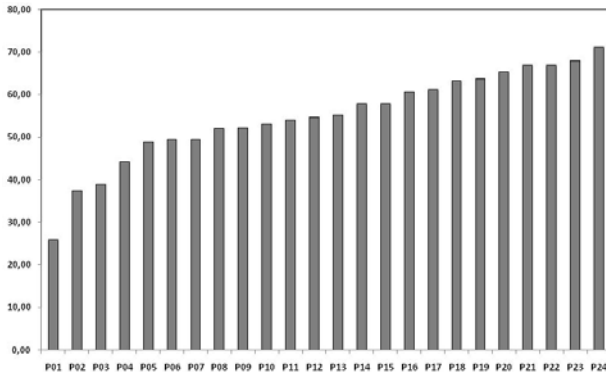


Fig. 1. Ordered classification accuracies of the participants

sort of genre similarities. Also, many genre pairs are never or extremely rarely confused, which implies that it is very easy for humans to distinguish these genres. Based on the user votes one can define the genre-song voting matrix $V = (v_{g,s})$, where $v_{g,s}$ denotes the number of times the participants voted for genre g given song s :

$$v_{gs} = \sum_{t \in T | s_t = s} \hat{g}_t == g \quad (2)$$

The genre-song voting matrix is visualized in figure 4. One can even visually see that the majority of the participants agree with the ground truth information for most of the songs. In contrast to the confusion matrix, the genre-song voting matrix visualizes the classification result for each song separately and is a compact representation of the results of the listening experiment. To further analyze the votes one can define the number of different genres $D(s)$ the participants have assigned to a specific song s :

$$D(s) = \sum_g^{G^+} v_{gs} > 0 \quad (3)$$

Figure 3 (left) shows a histogram of the number of different genres $D(s)$ the user voted for. Although there are 20 options to choose from, in general the participants did not vote for more than 8 different genres. This indicates that some genres are not relevant at all for some songs. Furthermore we can identify the most frequently estimated genre, the second most frequently estimated genre and so on, for each song. Then we can aggregate the number for votes for the k (1 to 20) most frequently estimated genre over all songs. The percentage of the accumulated votes relative to the total number of votes is visualized in figure 3 (right). Consistently with the histogram in figure 3 all votes are within the 12 most frequently estimated genres. In general there exists a strong consensus among the participants on a song's genre. The most frequently predicted genre

Alternative & Punk	59.2								5.0		0.4	0.4		2.1	2.9	0.4		30.4	0.4		0.4
Blues	2.5	46.3	0.4	0.4	0.8	13.3	1.3	0.4	0.8			15.4	0.4	1.3	4.2	2.9		6.7	0.4	1.7	0.8
Children's	0.8	0.4	51.2	4.2	2.1	3.3	1.7	1.7	1.7				2.1	0.4	12.1			10.0	0.8	5.8	3.3
Classical		0.8		66.3	0.4			2.9	0.8	9.2		0.8	3.8	0.8	4.2				4.6	1.3	5.0
Comedy&Spoken Word			0.8	1.3	91.7	0.4	0.4	0.4	0.4	0.4					1.7	0.4			2.1		
Country	0.4	2.9	0.8	0.4		56.7	1.7		2.9			0.4	0.4	7.9	0.8	0.4	7.1	0.8	16.3	2.1	
Easy Listening	0.8	2.9	0.4	5.0	0.8	2.5	32.1	1.3	0.4		4.2	0.8	23.8	8.3	1.7		7.1	4.2	1.3	3.3	
Electronic & Dance	0.8	0.4						94.6		0.8			0.4	0.4	0.4		1.7	0.4		0.4	
Vocals	0.4	0.4	1.3	0.8	0.4	17.1	5.8	0.4	5.0	0.4	0.4	1.3	1.7	10.8	0.4		6.3	1.7	39.6	5.8	
Hip-Hop									1.3	90.0				0.4		7.9	0.4				
Jazz		9.6		1.3			7.5				70.4	2.9	0.4	2.9	2.9			0.8		2.9	
Latin		0.8	0.8	0.8	0.4	12.1	0.8	0.8		5.4	45.8	0.8	6.3	1.3	0.4	6.3	1.7	5.0	14.2		
New Age	0.4	0.8		5.8	0.4	6.7	13.3	0.4			0.4	43.3	9.6				0.4	9.6			10.8
Other																					
R&B & Soul		1.3			0.4	2.9		2.9	2.1	1.3		0.8	2.9	77.5			7.1	0.4		0.8	
Reggae	0.4	0.8				0.4	4.2	1.3	0.8	18.8	0.4	3.8		0.4	5.0	57.1	4.2				3.3
Rock & Pop	6.3			0.4	0.4	2.1	1.7		2.9	1.3		3.3	5.0	6.3	0.4	67.6	1.3	0.8	1.3		
Soundtracks & More		0.4	0.8	12.1			6.7	5.0	0.4		0.8	1.3	15.8	8.3	0.4				45.8		2.9
Folk	0.4	1.7	2.1	8.3		2.1	5.4	0.4	33.3		0.8		1.3	15.0	9.2		11.3	2.1	3.3	5.0	
World			0.8	0.8	0.8	2.1	2.9	0.4	1.7		0.8	2.5	6.3	9.2	1.3	0.8	2.5	1.7	17.1	50.4	

Fig. 2. Confusion matrix of the classifications resulting from the experiment with respect to the ground truth annotation. Entry i, j is the percentage of user votes that predicted class j when the true class was i .

for each song is responsible for 64% of all votes. The two most frequently predicted genres of each song, together represent 80% of all votes (see figure 3). Therefore, we can conclude that the majority of the participants strongly agree on just one or two possible genre assignments for most of the songs.

3.2 Automatic Classification

To compare human to automatic classification performance we will use five different automatic classification methods. The choice of the evaluated approaches contains classical, well-known and state-of-the-art systems. Only complete genre classification systems as proposed in the literature are evaluated. Thus, the evaluated systems extract different feature sets and are based on different classification approaches. Two of the evaluated classification systems (SG-NN and RTBOF-NN) are based on nearest neighbor classifiers. The other three algorithms (GT-SVM, BLF1-SVM, BLF2-SVM) are based on a support vector machine classifier. The reported classification accuracies are obtained via leave-one-out cross-validation. The automatic classification methods are briefly described below.

Single Gaussian (SG-NN). The *Single Gaussian Nearest Neighbor Classifier* (SG-NN) is based on the so-called Bag of Frames (BOF) approach [1]. Each song is modeled as a distribution of Mel Frequency Cepstrum Coefficients (MFCCs). A single multivariate Gaussian distribution is used to model the distribution of MFCCs of a song. To identify the nearest neighbors the Kullback-Leibler (KL)

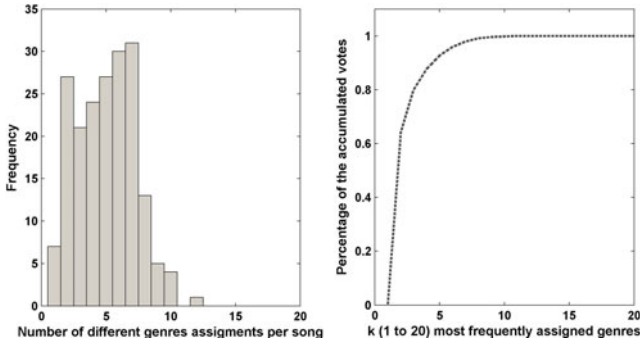


Fig. 3. Histogram of the number of different genres per song the participants have voted for (left) and percentage of the accumulated number of votes for the k most frequently assigned genres per song (right)

divergence between two models is computed. This approach is a fast and popular variant proposed by Levy et al. [9] of the classic timbre based audio similarity measure.

Rhythm Timbre Bag of Features (RTBOF-NN). The *Rhythm Timbre Bag of Features Nearest Neighbor Classifier* (RTBOF-NN) is a state-of-the-art music similarity measure proposed by Pohle et al. in [13]. This measure ranked first in the MIREX 2009 music similarity and retrieval task and has proven to be statistically significantly better than most of the participating algorithms. In contrast to the classic Single Gaussian approach this RTBOF-NN Classifier reflects the current state-of-the-art in nearest neighbor classification. Basically, it has two components – a rhythm and a timbre component. Each component, rhythm and timbre, consists of a distribution model over local spectral features. The features, described in [13], are complex and incorporate local temporal information over several frames. Because of its components we will call this approach Rhythm Timbre Bag Of Features (RTBOF) in our evaluations.

Block-Level Feature (BLF-SVM). The Block-Level Feature Support Vector Machine approach (BLF-SVM) is a genre classification algorithm based on block-level features. An earlier version of this algorithm [14] participated in the MIREX 2009 Audio Genre Classification task and took rank 14 out of 31. However, no statistically significant difference to the winning algorithm was found. This approach will be denoted BLF1-SVM. Additionally, we also evaluate an improved variant of this algorithm, which we call BLF2-SVM here. This algorithm includes three novel block-level features (Spectral Contrast Pattern, Correlation Pattern and Variance Delta Spectral Pattern). For a detailed description of these new feature we refer to [16]. This improved approach is expected to perform comparably to the state-of-the-art methods in genre classification.

Marsyas (MARSYAS-SVM). The Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) framework⁶ is an open source software that can be used to efficiently calculate various audio features. For a detailed description of the extracted features we refer to [19]. This algorithm has participated in the MIREX Genre Classification task from 2007 onwards, and the features as well as the classification approach have been the same over the years. We use the framework to extract the features exactly as for the MIREX contest (MARSYAS version 0.3.2). Then we use the WEKA Support Vector Machine implementation to perform cross-validation experiments. This method is closest to the automatic approach by Lippens et al. [10] and should help to make our experiment more comparable to this previous experiment.

3.3 Collaborative Classification

In this section we present two straight-forward collaborative classification approaches (CV and CSS-NN) based on the users' aggregated votes.

Collaborative Voting (CV). The Collaborative Voting (CV) approach is simple. The genre most participants have voted for is the predicted genre of a song. This method basically combines the individual classification results of the participants following the majority rule like a meta-classifier.

Collaborative Filtering (CF-NN). The Collaborative Filtering Nearest Neighbor Classifier (CF-NN) is related to an item-based collaborative filtering approach. Each song is represented by its voting profile, which corresponds to the column vector of a song in the genre-song voting matrix (see figure 4). One can then derive song similarities by comparing the voting profiles of the songs. To compare song profiles the *city-block* distance (l_1 norm) was used in our experiments. The song similarity information can then be used to perform nearest neighbor classification.

3.4 Comparison

In figure 5 the classification results of the automatic methods, the collaborative approaches and the individual results of the participants are visualized together, sorted according to the achieved accuracy. Clearly, the content-based approaches perform worse than most of the participants, whereas the collaborative approaches achieve high classification accuracies and outperform most of the participants. The observation that collaborative approaches do better than most individual humans can be explained by the fact that these type of algorithms better reflect the group opinion, which is the aggregated knowledge of many individuals. Therefore, the group as a whole has a broader musical knowledge than any individual, as each person is typically familiar with some but not with all genres of a classification dataset.

⁶ <http://marsyas.info>

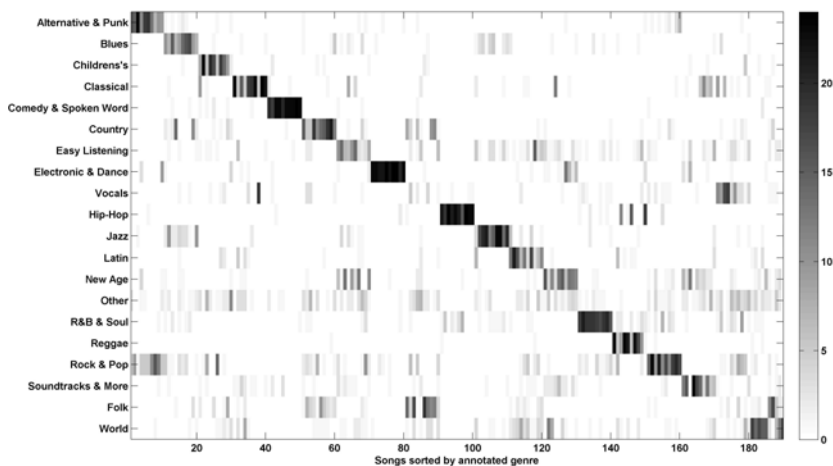


Fig. 4. Visualization of the genre-song voting matrix. Tracks are sorted according to the ground truth genre.

Comparing the best content-based approach (BLF2-SVM) to the best collaborative approach (CF-NN) it turns out that the latter achieves almost double the classification accuracy of the content-based approach. Taking a look at the various content-based method, we can see that there exist clear differences. The classical timbral similarity measure performs worst, just outperforming the worst participant. The classic MARSYAS-SVM approach does not perform much better, which slightly contradicted our expectations⁷ Both recent methods RTBOF and BLF2-SVM show an improvement in classification accuracy over the ‘classic’ approaches. This indicates that the improvements in automatic classification reduced the gap between human and automatic classification, but still there exists a difference of about 10 percentage points between the best automatic method and the average human participant. Furthermore, based on the obtained results we can define an upper bound on the achievable classification accuracy for automatic methods on this dataset. Clearly because of inconsistencies of the classification taxonomy and possible annotation errors none of the evaluated methods will ever reach perfect classification accuracy. However, as all evaluated methods have to deal with these problems the classification result of the CF-NN approach can be interpreted as an upper bound for automatic methods on this dataset.

4 Evaluation Based on User Data

One of the main disadvantages of using the classification accuracy as evaluation criterion is that such experiments heavily depend on the quality of the ground

⁷ Interestingly, when performing a 10-fold cross-validation instead of leave-one-out, we get comparable results for MARSYAS-SVM and BLF1-SVM. This effect is yet to be investigated.

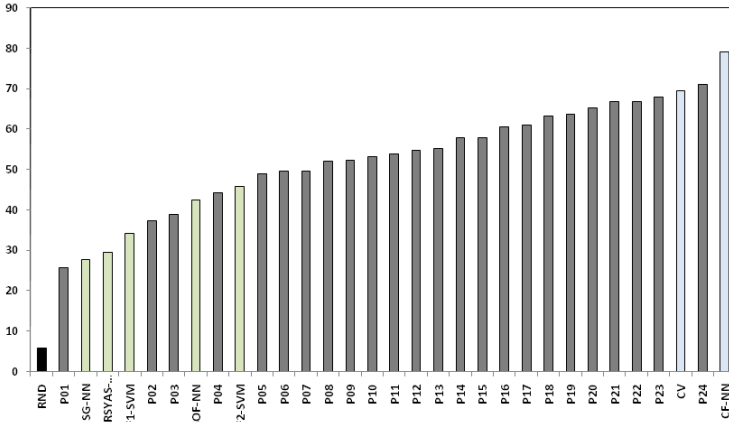


Fig. 5. Comparison of classification results of the individual participants, automatic methods and the collaborative approaches

truth annotations. To improve the quality of the ground truth one can of course ask an expert to define the genre annotations, but still the evaluation would just depend on a single opinion and as already pointed out there will always exist some annotation errors due to the inconsistency of the genre taxonomy itself.

To overcome these limitations we propose to perform a user centric evaluation by aggregating the collected genre votes of the participants of the listening experiment. Thus, the ground truth is no longer based on a single opinion, but on the aggregated opinions of all the participants regarding the genre affinity of a given song. This way we can not only use the obtained data from the listening experiment to make automatic classification methods comparable to human classification performance, but this information can also be used to account for genre ambiguities whenever genre classification is used in an evaluation, as already proposed in [3] and [10]. The basic idea for such a quality measure is straight-forward: If even humans are unsure about a genre label then it will be hard for the machine to get the label right.

To reflect these uncertainties of the genre annotations in a quality measure, a *user score* is defined similarly to [10]. A user score measures the agreement of the predictions of an automatic method with the genre assignments of the humans participating in the listening experiment. Thus, any algorithm can collect points for each song s in the dataset according to the agreement with the user votes. In particular, for each song $s \in S$ the classification of the algorithm into genre $\hat{g}_s \in G$ is rated by the number of times this genre was voted for ($v_{\hat{g}_s, s}$) relative to the number of times the participants voted for the most frequently predicted genre ($\max(\{v_{g, s} | g \in G\})$).

$$\text{US1} = \frac{1}{|S|} \sum_s^{s \in S} v_{\hat{g}_s, s} / \max(\{v_{g, s} | g \in G\}) \quad (4)$$

Extending the idea in [3], another straight-forward definition of a *user score* — this score is denoted US2 — is to take the number of collected points relative to the maximum number of points one can obtain on the dataset.

$$\text{US2} = \sum_s \frac{v_{\hat{g}_s, s}}{\sum_s \max(\{v_{g, s} | g \in G\})} \quad (5)$$

The difference of the two scores is that for US1 each song contributes equally, whereas for US2 it is more important to correctly predict songs where the participants agreed pretty much on a single genre. One important advantage of both user scores is that they no longer rely on the ground truth annotation, but are solely based on the user ratings. By definition both scores are in the range between 0 and 1.

Table 2. Comparison of the user scores (US1, US2) and the classification accuracy (acc.) obtained for the automatic approaches presented in section 3.2

Approach	US1	US2	acc.
BLF2-SVM	0.5615	0.5080	0.4579
RTBOF-NN	0.4352	0.3827	0.4253
BLF1-SVM	0.3672	0.3382	0.3421
MARSYAS-SVM	0.3217	0.3031	0.2953
SG-NN	0.3156	0.2791	0.2779
RND	0.0578	0.0673	0.0584

Table 2 summarizes the user scores and the classification accuracy for the automatic classification methods presented in section 3.2. To our knowledge this is the **first comparison** of automatic classification methods also accounting for genre ambiguities in the literature. The ranking of the analyzed algorithms is the same for all quality criteria. However, taking genre ambiguities into account clearly changes the evaluation result. For example the difference between the BLF2-SVM and the RTBOF-NN is relatively bigger for the users scores compared to the classification accuracy. An improvement of a user score over the classification accuracy reveals that the misclassified songs are not classified into an arbitrary, completely unrelated genre, but into a genre that users find similar, or tend to confuse also. We advocate this method for future evaluations of genre classifiers, whenever appropriate data are available.

5 Conclusions

Based on the evaluation results presented in section 3.4, we can conclude that there is some progress with respect to automatic genre classification methods, reducing the gap between automatic methods and human classification. However, the best performing automatic method in our experiment still performs about

10 percentage points worse than the average human participant. Furthermore, we could also show that the collaborative approach outperforms both automatic methods as well as individual human performances. Thus, collaboratively collecting meta-information about music e.g. via a music platform is a very powerful method and is also the clear trend in the music business. For content-based methods this implies that they are only beneficial in situations where no other data is available – for instance, in cold start situations, or in special application scenarios where no access to collaboratively collected meta-data is possible. Additionally, with respect to the evaluation of content-based systems we have proposed two user centric evaluation criteria. The proposed user-scores no longer depend on a single ground truth annotation, but on the aggregate opinion of the participants of the conducted listening experiment. One advantage of the proposed user-scores is that they account for genre ambiguities which will help to improve the evaluation of automatic classification systems in future, especially since the whole dataset (including both the audio files and the collected votes) is publicly available.

Acknowledgments. This research was supported by the Austrian Research Fund (FWF) under grant L511-N15. We especially thank all the participants of the listening experiment. It took many of them hours to complete the survey.

References

1. Aucouturier, J.J., Defreville, B., Pachet, F.: The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America* (2007)
2. Bella, S.D., Peretz, I.: Differentiation of classical music requires little learning but rhythm. *Cognition* (2005)
3. Craft, A., Wiggins, G.A., Crawford, T.: How many beans make five? the consensus problem in music-genre classification and a new evaluation method for single-genre categorisation systems. In: *Proc. Int. Sym. on Music Information Retrieval, ISMIR 2007* (2007)
4. Ellis, D., Whitman, B., Berenzweig, A., Lawrence, S.: The quest for ground truth in musical artist similarity. In: *Proc. of the 3rd International Conference on Music Information Retrieval, ISMIR 2002* (2002)
5. Flexer, A., Schnitzer, D.: Album and artist effects for audio similarity at the scale of the web. In: *Proc. of the 6th Sound and Music Computing Conference, SMC 2009* (2009)
6. Geleijnse, G., Schedl, M., Knees, P.: The quest for ground truth in musical artist tagging in the social web era. In: *Proc. of the 8th International Conference on Music Information Retrieval, ISMIR 2007* (2007)
7. Gjerdingen, R., Perrott, D.: Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research* (2008)
8. Guaus, E., Herrera, P.: Music genre categorization in humans and machines. In: *121 AES Convention* (2006)
9. Levy, M., Sandler, M.: Lightweight measures for timbral similarity of musical audio. In: *AMCMM 2006: Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia, Santa Barbara, California, USA, pp. 27–36* (2006)

10. Lippens, S., Martens, J., Mulder, T.D., Tzanetakis, G.: A comparison of human and automatic musical genre classification. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2004 (2004)
11. McKay, C., Fujinaga, I.: Musical genre classification: Is it worth pursuing and how can it be improved? In: Proc. of the 7th Int. Conf. on Music Information Retrieval, ISMIR 2006 (2006)
12. Pachet, F., Cazaly, D.: A taxonomy of musical genre. In: Proc. of Content-Based Multimedia Information Access Conference, RIOA (2000)
13. Pohle, T., Schnitzer, D., Schedl, M., Knees, P., Widmer, G.: On rhythm and general music similarity. In: Proc. of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009 (2009)
14. Seyerlehner, K., Schedl, M.: Block-level audio feature for music genre classification. In: Online Proc. of the 5th Annual Music Information Retrieval Evaluation eXchange, MIREX 2009 (2009)
15. Seyerlehner, K., Widmer, G., Knees, P.: Frame level audio similarity - a codebook approach. In: Proc. of the 11th International Conference on Digital Audio Effects, DAFx 2008 (2008)
16. Seyerlehner, K., Widmer, G., Pohle, T.: Fusing block-level features for music similarity estimation. In: Proc. of the 13th International Conference on Digital Audio Effects, DAFx 2010 (2010)
17. Soltau, H., Schultz, T., Westphal, M., Waibel, A.: Recognition of music types. In: Proc. of the 23rd IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 1998 (2010)
18. Sordo, M., Celma, O., Blech, M., Guaus, E.: The quest for musical genres: Do the experts and the wisdom of crowds agree? In: Proc. of the 9th International Conference on Music Information Retrieval, ISMIR 2008 (2008)
19. Tzanetakis, G., Cook, P.: Musical genre classification of audio signal. IEEE Transactions on Audio and Speech Processing (2002)

Clubmixer: A Presentation Platform for MIR Projects

Alexander Schindler^{1,2} and Andreas Rauber¹

¹ Department of Software Technology and Interactive Systems
Vienna University of Technology

{schindler,rauber}@ifs.tuwien.ac.at

² Business Unit High Performance Image Processing
Department Safety & Security
Austrian Institute of Technology

Abstract. Evaluating solutions to many music IR problems – such as playlist generation, music similarity – in absence of formal evaluation measures frequently requires user studies to establish the benefits of one solution over the other. Building an according application framework to deploy and test user responses is a cumbersome and complex task. We present Clubmixer - an advanced client-server based audio system that could serve MIR researchers as presentation and prototyping platform. The project aims at providing a software framework that minimizes the effort of creating MIR based solutions. The open architecture and the use of open standards provide high flexibility for several MIR related areas (e.g. content based retrieval, collaborative retrieval, etc.). We describe the current state of the system and outline the main functionality as well as the advantages of Clubmixer for MIR research.

Keywords: MIR systems and infrastructure, user interfaces and music access.

1 Introduction

Although the discipline of music information retrieval (MIR) has matured since the early 1990s, MIR technology is not yet as widely used as research would like to see it. One part of the challenge may lie in the gap between the availability of sophisticated algorithms and research results in the prototype stage that promise superior performance and advanced features, and the evaluation in how far these promises live up to their expectations and meet user demands. The challenge, in most cases, lies in the fact that the approaches resulting from sophisticated research need to be deployed within a real system environment offering a rather large number of – by now standard – features expected by users in addition to the functionality offered by the research prototype. Building and deploying such a complex system constitutes a significant challenge on its own, putting a significant burden on researchers in music IR.

Examples of publicly available MIR solutions are still rather limited. Nowadays computers already have the appropriate resources to analyze average sized

private music collections with state of the art MIR technologies - Yet there are virtually no software audio players, nor plugins for commonly available software implementing MIR technology, despite the existence of a significantly large number of research prototypes.

To gather broader acceptance and recognition outside the research community, new solutions have to be presented and made available in a commonly acceptable form. Prototypes should provide user interfaces that correspond to the look and feel of commonly available audio software.

To meet these goals we present *Clubmixer*, a cross platform client-server audio jukebox system that can serve as a presentation platform for MIR research prototypes. It offers a number of features that are expected by users as default requirements, both on the functional as well as user interface level. Combined with a flexible architecture, existing MIR solutions can be plugged in, offering a sophisticated basis for the evaluation and deployment of MIR solutions in a setting acceptable by consumers.

The remainder of this paper is organized as follows: Section 2 presents related work on MIR systems, Section 3 describes the Clubmixer system followed by some example scenarios how to implement MIR solutions in Clubmixer in Section 4. Section 5 presents our conclusions.

2 Related Work

A good summary of music information retrieval systems is presented by Typke et. al [13]. Several prototyping frameworks have been introduced, like the well known C++ software framework CLAM [1]. It offers tools and repositories, as well as visual components, which can be used to rapidly develop research prototypes in the audio and music domain. The rapid prototyping environment Chuck [4] is a high-level programming language for music and sound synthesis including content analyzing and learning frameworks. Jmir [9] is a free and open-source software suite for automatic music classification, including audio, symbolic and Web content feature extractors. These projects generally focus on providing algorithmic components, whereas Clubmixer aims at providing a representative user interface combined with an open framework where further solutions can be easily integrated. Kurth et. al [6] presented SyncPlayer - a client-server based framework for multimodal presentation of audio and associated music-related data, which is conceptually similar to Clubmixer. The multiuser concept, used by Clubmixer was also introduced by [5] and [12] for collaborative playlist generation. These projects focus on collaborative balanced playlist generation with little or no use of content based retrieval techniques.

Songbird [1] is a cross-platform media player built on the Mozilla application framework. [8] gives a brief introduction into Songbird and details how to write add-ons by the example of the automatic playlist generation and music library visualization add-on *Soundbite for Songbird*.

¹ <http://www.getsongbird.com/>

3 Clubmixer Framework

The Clubmixer framework is a cross platform audio jukebox system based on a client-server architecture. The chosen architecture provides a major advantage for common MIR related tasks that largely depend on processing intensive calculations or time consuming feature extractions. Clubmixer facilitates a distributed execution where resource intensive calculations can be carried out on computational adequate server machines. User interactions, presentations or evaluations can be performed remotely over the network utilizing the provided client user interface. The framework is aligned to the look and feel of currently available software audio players. This benefits rapid prototyping. Especially projects targeting the optimization of algorithms often do not require a full featured user interface. Consequently, this time intensive task is often neglected and results are presented by command line tools.

The Clubmixer framework integrates the Java Plugin Framework (JPF) and provides several points, where the core software components can be extended by plugins. MIR related prototypes can use these extension points to tailor the framework to their needs. Additionally a set of standard components is provided that can be reused within plugins to reduce the development effort and maintain a common look and feel. Clubmixer is based on standard Web technologies and protocols, which provides a high degree of flexibility for researchers in building or integrating client solutions on nearly every platform (e.g. mobile devices, Web pages). Further the possibility to spawn multiple distributed clients for a single server instance accounts for research areas related to collaborative information retrieval.

Music Information Retrieval is highly dependent on metadata that is extracted from multiple sources. It is common practice to store this data in semistructured text-files which restricts the possibility of adapted queries and analyses. Clubmixer provides and automatically extracts a predefined set of track metadata and uses a solid database system for data storage. MIR prototypes can access and extend the initial database schema and store their extracted metadata directly in the media library. Though there are several Java implementations of audio content extraction algorithms [9,7], Clubmixer is not limited to Java based solutions only. Clubmixer provides a console mode with a predefined command set and full database access. This set can be extended and provides a convenient way to write small commands that are executed only occasionally and don't need a full integration into the framework.

3.1 Contributions

This section provides a brief overview of related work and examples, that can be quickly implemented using the Clubmixer framework:

- Preference based automatic playlist generation systems in shared environments like Adaptive Radio [3] or PartyVote [12]. The Clubmixer framework already provides all necessary functions (including user management). Only

UI elements and data descriptions for additional preferences have to be implemented.

- Jukola [10], a field trial of a collaborative playlist generation system, using multiple mobile devices and static touch screen panels where users can interact with the system. Due to the open Web service interfaces of Clubmixer, clients for mobile devices can be developed conveniently.
- Relevance feedback based systems only need to implement a set of additional buttons as well as routines for data storage/retrieval.
- Content based feature extraction can be integrated directly through Java implementations. External extracted features can be imported into the database using the console mode with customized import scripts.
- User evaluations to assess the quality of automatically generated results (e.g. automatically generated playlists, music recommendations, genre classification, etc.). Clubmixer client can be extended to evaluate the results or a new client with reduced functionality can be implemented quickly.

3.2 Clubmixer Server

Clubmixer Server is the main component of the framework. It provides all features of a standard software audio player or jukebox system. Digital audio files² have to be locally available on the hosting computer. The files are automatically imported from user-specified directories and the extracted metadata is stored in the media library. The provided import routines can be extended to extract any kind of metadata that is needed for a custom MIR prototype (see Section 3.2).

To come up to the computational requirements of MIR research prototypes, especially when sophisticated signal processing and machine learning algorithms are utilized, Clubmixer server could be hosted on a high-performance computer. The application hides to the system tray but it can even be run in a headless terminal. This provides an advantage over other software audio players that cannot be executed on hosts where only shell access is provided. Only for launching the configuration window (see Fig. 1) a window manager is needed.

The following sections describe the components of Clubmixer Server. Some of these components provide extension points - defined program sections that can be functionally extended by plugins. Brief descriptions of these points as well as their benefits for MIR research are given.

Data Storage. Clubmixer uses a Hibernate³ persistence layer with a HyperSQL⁴ (HSQL) database as data storage. This overcomes the commonly reported performance degradation on comparable audio software with huge song collections [2]. The intermediate persistence layer additionally provides the advantage to exchange the underlying database system and to extend the initial database

² Currently only MP3 and WAV audio are fully supported.

³ <http://www.hibernate.org>

⁴ <http://hsqldb.org>

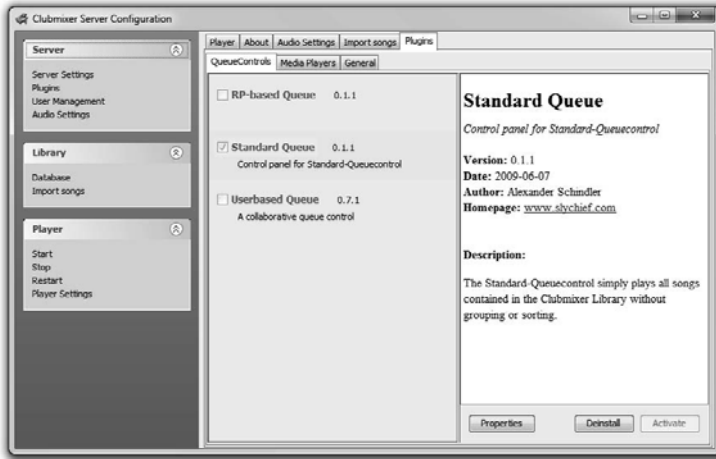


Fig. 1. Clubmixer Server Configuration Window

scheme by plugins. This gives developers the opportunity to store their data directly in the applications database.

Configuration properties (e.g. database connection settings) are stored separately utilizing the Java Preferences API, which stores these properties according to the underlying operating system (Windows Registry on Windows systems, config-files on Linux systems).

Communication. Communication is based on standard Web service technology. Three communication channels (see Fig. 2) are implemented as SOAP Web services. The first Web service provides standard audio player functionality (play, stop, next, add to playlist, etc.). The library service provides an extended search interface with filters on multiple song attributes (queries are wildcarded to enable searching for keywords).

A third Web service has been introduced to provide a generic communication port for plugins. Due to the distributed architecture of Clubmixer, plugins too consist of two separated parts - a client and a server part - which need to exchange data and invoke methods. A custom Plugin Communication Channel (PCC) - a lightweight distributed middleware - provides a very flexible interface, where the client-component of a plugin can invoke methods of the server-component (even if the client side has been implemented in a different programming language). The standard invocation of remote server-methods requires clients to wrap primitive parameters into sets of key-value pairs (e.g. in a HashTable). This approach provides best compatibility for non-Java clients. A special generic method invocation is accessible if the parameters of the remote methods are JAXB⁵

⁵ <http://jaxb.dev.java.net/>

annotated data objects. This method is more convenient and allows to use more complex objects as method parameters, but restricts development to the Java programming language. Further protocols (e.g. REST, JSON) are currently not supported, but can easily be added through plugins.

State change events of the server (song change, playlist change) are propagated through an Apache ActiveMQ⁶ message queue. Clients can subscribe to this queue and invoke Web service methods to synchronize their states accordingly.

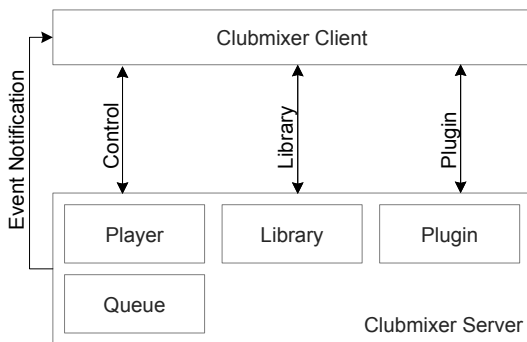


Fig. 2. Clubmixer Communication Channels

User Management. Clubmixer is a multiuser system and integrates user management with access controls. Users can be assigned several access rights (e.g. start/stop the server, skip songs, etc.). This could offer a guest in a bar the possibility to search and enqueue a certain song but prevent him from stopping the server. An integrated user management enables plugins to store user preferences. This can be addressed by MIR topics like playlist generation, collaborative filtering and relevance feedback.

Extension Points. Extension points are predefined points of the application, where the core functionality can be extended by plugins. Currently the following extension points are implemented:

- **General** - general functionality (e.g. opening sockets, running tasks)
- **Importer** - if the plugin requires additional data (e.g. audio feature vectors), a custom importer can be added.
- **Persistence** - extends the standard database schema by simply providing further JPA⁷ annotated entity classes.

⁶ <http://activemq.apache.org/>

⁷ Java Persistence API,

<http://java.sun.com/javaee/technologies/persistence.jsp>

- **Queue Control** - A queue control is an extended automatic playlist generator that provides a constant queue of songs.
- **Console** - extends the standard command set of the console mode

A player extension point to exchange the current Java based player with native audio player implementations is planned. This will provide more flexibility and will overcome known issues concerning the Java audio libraries (Javalayer⁸ and Tritonus⁹).

Annotation based dependency injection is used to provide all necessary functionality within plugins (e.g. database access, player control, etc.). Listing 1.1 gives an example of how to use annotations to get the references to the required components.

Console Mode. The console mode provides partially access to functions and services of Clubmixer server without the requirement of a fully running system. It further implements a set of commands to invoke certain parts of the server (e.g. loading plugins into the environment, starting the database). This command set can be easily expanded - the provided interface invokes the commands and passes on the supplied parameters - similar to standard main-methods of common programming languages.

3.3 Clubmixer Client

A Clubmixer Client acts only as a front-end to the server. It can be used to control the server and search for songs in the library. The aim is to provide a user interface that implements the average look and feel of currently available audio software. The default Clubmixer Client is a Java Swing client (see Fig. 3) which provides commonly known features of an audio software player. It can be used to control the playback of recordings, query for songs, manipulate playlists and display additional information about songs and artists.

To enable fast development of client side plugins and to provide a common look and feel, several GUI elements are provided in a custom GUI components library - the *Common GUI Elements*. It provides among others, components to display song metadata with an albumart image, popup menus and diverse event handlers. Clubmixer Client currently provides three extension points, that can be used by plugins to add components for displaying data or to trigger server side methods.

Creating a Custom Client. Due to the open standards and libraries (SOAP, ActiveMQ) clients for Clubmixer Server can be implemented on almost every platform in almost any programming language. There are already initial implementation for Windows Mobile and JavaME. Plugin projects that are intended to provide information to non-Java clients should refrain from using complex data types as method parameter, due to the constraints described in Section 3.2

⁸ <http://www.javazoom.net/javayer/javaayer.html>

⁹ <http://www.tritonius.org>

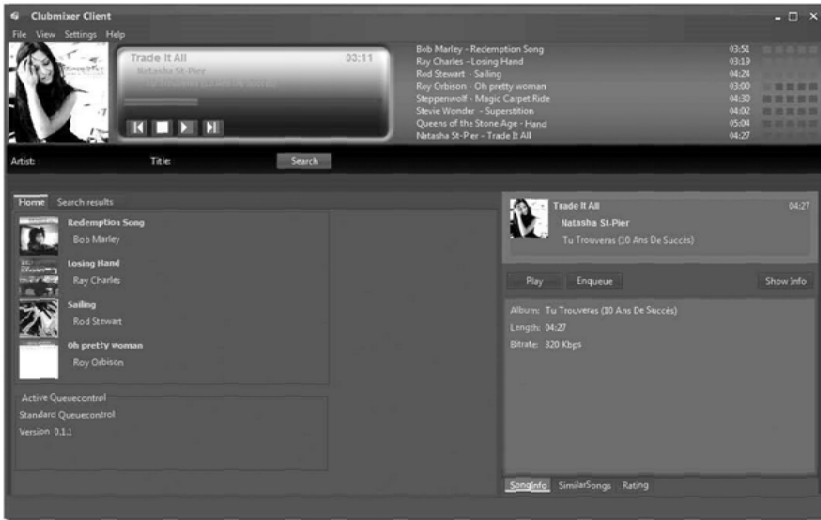


Fig. 3. Clubmixer Client

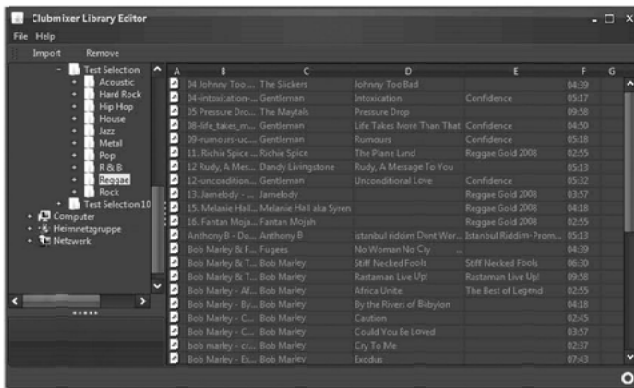


Fig. 4. Clubmixer Library Editor

3.4 Clubmixer Library Editor

Clubmixer Library Editor (see Fig. 4) is intended to be the central place for querying and editing every information that is stored in the library. It provides a file system browser and displays imported metadata for MP3 files.

4 Creating a New MIR Based Clubmixer Plugin by Example

This section gives a brief overview of how to implement MIR projects as Clubmixer plugins. The exemplified scenario describes the common task of calculating song similarities. A content based solution has been applied which requires a plugin that takes advantage of several extension points.

To provide content based similarity calculations, further information has to be extracted from the audio files. Thus, the plugin has to provide a custom importer that extracts feature vectors and calculates the similarity matrices. To store this data efficiently, the the preexisting database schema has to be extended by a set of new entity classes.

Listing 1.1 shows an example implementation of the fully operational plugin. It provides database access as well as a storage container for configuration properties.

```

1
2 public class MirPlugin extends Plugin {
3
4     @CommunicationChannel(pluginname = "MirPlugin")
5     private ICommunicationChannel com;
6
7     @ServerLibrary
8     private ClubmixerServerLibrary lib;
9
10    @Preferences
11    private ClubmixerPreferences prefs;
12
13    public MirPlugin() {
14
15    }
16
17    public List<Song> findSimilar(Song s) {
18
19        // MIR based algorithms
20        ...
21
22    }
23
24 }
```

Listing 1.1. Example Plugin Implementation

The previous two paragraphs outlined all relevant code that has to be implemented at the server side. In order to display the extracted results, client side extension points have to be addressed. Fig. 5 a) shows a standard popup menu that is provided by the Common GUI Elements library. This popup menu can be linked to several song-related components and offers standard actions for the related song. It can be easily extended by adding further menu items. Fig. 5 b) shows the menu extended by the entry 'find similar' which offers to search for songs similar to the selected one. Listing 1.2 depicts the entire source code for this extension. The custom menu uses the Plugin Communication Channel to invoke the server side method 'findSimilar' from Listing 1.1, which processes the request and returns a list of similar songs. The retrieved result is passed on to

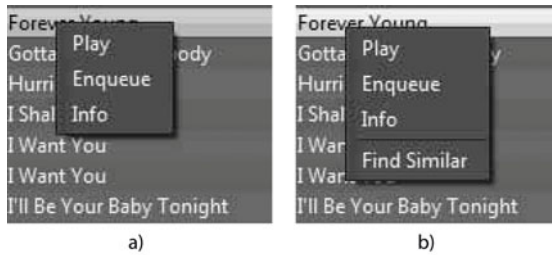


Fig. 5. Image a) shows the standard popup menu that is provided by the Commons GUI Elements library. Image b) shows the menu with an additional item that has been added by a plugin.

a client component which is responsible for updating and displaying the search result table.

```

1
2
3 public class MenuItem extends JMenuItem implements IMenuSong {
4
5     @CommunicationChannel(pluginname = "MirPlugin")
6     private ICommunicationChannel com;
7
8     @SearchresultHandler
9     private SearchResultHandler srh;
10
11
12     private Song currentSong;
13
14     public MenuItem() {
15
16         ActionMap map = ApplicationContext.getActionMap();
17         this.setAction(map.get("getSimilarSongs"));
18         this.setText("Find Similar");
19     }
20
21     @Override // from interface IMenuSong
22     public void setSong(Song song) {
23         this.currentSong = song;
24     }
25
26     @Action
27     public Task getSimilarSongs() {
28
29         // get reference to server method
30         GenericRemoteMethod<List<Song>, Song> findSimilar =
31             com.getGenericRemoteMethod("findSimilar");
32
33         // invoke remote method
34         List<Song> similarSongs = findSimilar.invoke(currentSong);
35
36         // output result list
37         srh.fireSearchResultChanged(similarSongs);
38     }
39 }
40

```

Listing 1.2. Extending the Standard Popup Menu

5 Conclusion and Future Work

The proposed Clubmixer framework is a solid and easy to extend audio player software, that has been developed in consideration of being used for music information retrieval research. It combines a good architecture with an appealing graphical user interface.

We have demonstrated how a MIR project can be turned into a Clubmixer plugin by only a few steps. Thus, new algorithms can be presented in an audio framework that incorporates the standard look and feel of currently available audio software.

There are various areas of applications that are currently being investigated and evaluated:

- Combining the SOMEjB Music Digital Library Project [11] and Clubmixer by integrating SOMEjB as a customized plugin.
- Extending the Library Editor with analytical functions to statistically analyze extracted audio features.
- A Matlab connector to control Clubmixer from within the Matlab environment as well as to exchange data.

6 Software and Source Code

Clubmixer is hosted as SourceForge project. Software installer packages and project source code can be found at <http://sourceforge.net/projects/clubmixer/>.

References

1. Amatriain, X., Arumi, P., Garcia, D.: A framework for efficient and rapid development of cross-platform audio applications. *Multimedia Systems* 14(1), 15–32 (2008)
2. Byfield, B.: Comparing five music players. *Linux Journal* 193(4) (2010)
3. Chao, D., Balthrop, J., Forrest, S.: Adaptive radio: achieving consensus using negative preferences. In: *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work*, New York, NY, pp. 120–123 (2005)
4. Fiebrink, R., Wang, G., Cook, P.: Support for mir prototyping and real-time applications in the chuck programming language. In: *9th International Conference on Music Information Retrieval* (2008)
5. O’Hara, K., Lipson, M., Jansen, M., Unger, A., Jeffries, H., Macer, P.: Jukola: democratic music choice in a public space. In: *DIS 2004: Proceedings of the 5th Conference on Designing Interactive Systems*, pp. 145–154. ACM, New York (2004)
6. Kurth, F., Müller, M., Damm, D., Fremerey, C., Ribbrock, A., Clausen, M.: Syncplayer - an advanced system for multimodal music access. In: *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, London, UK, pp. 381–388 (2005)
7. Lidy, T., Rauber, A.: Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In: *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, London, UK, pp. 34–41 (2005)

8. Lloyd, S.: Automatic playlist generation and music library visualisation with timbral similarity measures. Master's thesis, Queen Mary University of London (August 2009)
9. McKay, C., Fujinaga, I.: jmir: Tools for automatic music classification. In: Proceedings of the International Computer Music Conference (2009)
10. OHara, K., Lipson, M., Jansen, M., et al.: Jukola: democratic music choice in a public space. In: Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, Cambridge, MA, USA, pp. 145–154 (2004)
11. Rauber, A., Pampalk, E., Merkl, W.: The SOM-enhanced JukeBox: Organization and Visualization of Music Collections based on Perceptual Models. *Journal of New Music Research*, 193–210 (2003)
12. Sprague, D., Wu, F., Tory, M.: Music selection using the partyvote democratic jukebox. In: AVI 2008: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 433–436. ACM, New York (2008)
13. Typke, R., Wiering, F., Veltkamp, R.C.: A survey of music information retrieval systems. In: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005), London, UK, pp. 153–160 (2005)

Similarity Adaptation in an Exploratory Retrieval Scenario

Sebastian Stober and Andreas Nürnberger

Data & Knowledge Engineering Group
Faculty of Computer Science

Otto-von-Guericke-University Magdeburg, D-39106 Magdeburg, Germany
{Sebastian.Stober,Andreas.Nuernberger}@ovgu.de

Abstract. Sometimes users of a multimedia retrieval system are not able to explicitly state their information need. They rather want to browse a collection in order to get an overview and to discover interesting content. Exploratory retrieval tools support users in search scenarios where the retrieval goal cannot be stated explicitly as a query or user rather want to browse a collection in order to get an overview and to discover interesting content. In previous work, we have presented Adaptive SpringLens – an interactive visualization technique building upon popular neighborhood-preserving projections of multimedia collections. It uses a complex multi-focus fish-eye distortion of a projection to visualize neighborhood that is automatically adapted to the user’s current focus of interest. This paper investigates how far knowledge about the retrieval task collected during interaction can be used to adapt the underlying similarity measure that defines the neighborhoods.

1 Introduction

Growing collections of multimedia data such as images and music require new approaches for exploring a collection’s contents. A lot of research in the field of multimedia information retrieval focuses on queries posed as text, by example (e.g. query by humming and query by visual example) as well as automatic tagging and categorization. These approaches, however, have a major drawback – they require the user to be able to formulate a query which can be difficult when the retrieval goal cannot be clearly defined. Finding photos that nicely outline your latest vacation for a presentation to your friends is such a retrieval goal and underlining the presentation by a suitable background music cannot be done with query by example. In such a case, exploratory retrieval systems can help by providing an overview of the collection and let the user decide which regions to explore further.

When it comes to get an overview of a collection, neighborhood-preserving projection techniques have become increasingly popular. Beforehand, the objects to be projected have to be analyzed to extract a set of descriptive features. (Alternatively, feature information may also be annotated manually or collected from external sources.) Based on these features, the objects can be



Fig. 1. Galaxy user-interface visualizing a photo collection with an object marked green in primary focus and two objects in secondary focus. (color scheme inverted for better printing).

compared – or more specifically: appropriate distance- or similarity measures can be defined. The general objective of the projection can then be paraphrased as follows: Arrange the objects (on the display) in such a way that neighboring objects are very similar and the similarity decreases with increasing object distance (on the display). As the feature space of the objects to be projected usually has far more dimensions than the display space, the projection inevitably causes some loss of information – irrespective of which dimensionality reduction techniques is applied. Consequently, this leads to a distorted display of the neighborhoods such that some objects will appear closer than they actually are, and on the other hand some objects that are distant in the projection may in fact be neighbors in feature space.

In previous work [10,13], we have developed an interface for exploring image and music collections using a galaxy metaphor that addresses this problem of distorted neighborhoods. Figure 1 shows a screenshot of the interface visualizing a photo collection. Each object is displayed as a star (i.e. a point) with its brightness and (to some extent) its hue depending on a predefined importance measure – e.g. a (user) rating or a view / play count. A spatially well distributed subset of the collection (specified by filters) is additionally displayed as a small image (a thumbnail or album cover respectively) for orientation. The arrangement of the stars is computed using multi-dimensional scaling (MDS) [5] relying on a set of descriptive features to be extracted beforehand. (Alternatively, feature

information may also be annotated manually or collected from external sources.) MDS is a popular neighborhood-preserving projection technique that attempts to preserve the distances (dissimilarities) between the objects in the projection. The result of the MDS is optimal w.r.t. the minimization of the overall distance distortions. Thus, fixing one distorted neighborhood is not possible without damaging others. However, if the user shows interest in a specific neighborhood, this one can get a higher priority and be temporarily fixed (to some extent) at the cost of the other neighborhoods. To this end, an adaptive distortion technique called SpringLens [4] is applied that is guided by the user’s focus of interest. The SpringLens is a complex overlay of multiple fish-eye lenses divided into primary and secondary focus. The primary focus is a single large fish-eye lens used to zoom into regions of interest compacting the surrounding space but not hiding it from the user to preserve overview. While the user can control the primary focus, the secondary focus is automatically adapted. It consists of a varying number of smaller fish-eye lenses. When the primary focus changes, a neighbor index is queried with the object closest to the center of focus. If nearest neighbors are returned that are not in the primary focus, secondary lenses are added at the respective positions. As a result, the overall distortion of the visualization temporarily brings the distant nearest neighbors back closer to the focused region of interest. This way, distorted distances introduced by the projection can to some extent be compensated.

The user-interface has been evaluated in a study as reported in [9]. In the study, 30 participants had to solve an exploratory image retrieval task: Each participant was asked to find representative images for five non-overlapping topics in a collection containing 350 photographs. This was repeated on three different collections – each one with different topics (and with varying possibilities for interaction). The evaluation showed that the participants indeed frequently used the secondary focus to find other photos belonging to the same topic as the one in primary focus. However, some photos in secondary focus did not belong to the same topic. Thus, this paper aims to answer the question whether it is possible to automatically adapt the neighborhood index during the exploratory search process to return more relevant photos for the primary focus topic.

The remaining paper is structured as follows: Section 2 outlines the experimental setup comprising the datasets, features and the definition of the distance facets. The adaptation method is covered by Section 3. The experiments are described in Sections 4 to 6. Section 7 draws conclusions.

2 Experimental Setup

2.1 Dataset

Four image collection were used during the study of which the first one (Melbourne & Victoria) is not considered here because it was only used for the introduction of the user-interface and has no topic annotations. All collections

Table 1. Annotated Photo collections and topics used in the user study

collection	topics (number of images)
Barcelona	Tibidabo (12), Sagrada Família (31), Stone Hallway in Park Güell (13), Beach & Sea (29), Casa Milà (16)
Japan	Owls (10), Torii (8), Paintings (8), Osaka Aquarium (19), Traditional Clothing (35)
Western Australia	Lizards (17), Aboriginal Art (9), Plants (Macro) (17), Birds (21), Ningaloo Reef (19)

were drawn from a personal photo collection of the authors¹. Each annotated collection comprises 350 images scaled down to fit 600x600 pixels – each one belonging to at most one of five non-overlapping topics. [Table 1](#) shows the topics for each collection. In total, 264 of the 1050 images belong to one of the 15 topics.

2.2 Features

For all images the MPEG-7 visual descriptors EdgeHistogram (EHD), Scalable-Color (SCD) and ColorLayout (CLD) [\[8\]](#) were extracted using the Java implementation provided by the Caliph&Emir MPEG-7 photo annotation and retrieval framework [\[6\]](#).

The EHD captures spatial distributions of edges in an image. The images are divided into 4×4 sub-images. Using standard edge detection methods, the following 5 edge types are detected: vertical, horizontal, 45° , 135° and non-directional edges. The frequency of these edge types is stored for each sub-image resulting in 16×5 local histogram bins. Further, a global-edge histogram (5 bins) and 13 semiglobal-edge histograms (13×5 bins) are directly computed from the local bins. The 13 semiglobal-edge histograms are obtained through grouping 4 vertical sub-images (4 columns), 4 horizontal sub-images (4 rows) and 4 neighbor sub-images (5 (2×2)-neighborhoods).

The SCD is based on a color histogram in the HSV color space with a fixed color space quantization. Coefficients are encoded using a Haar transform to increase the storage efficiency. Here, we use 64 coefficients which is equivalent to 8 bins for the hue (H) component and 2 bins each for the saturation (S) and the value (V) in the HSV histogram.

The CLD is also based on color histograms but describes localized color distributions of an image. The image is partitioned into 8×8 blocks and the average color is extracted on each block. The resulting iconic 8×8 “pixel” representation of the image is expressed in YCrCb color space. Each of the components (Y, Cr, Cb) is transformed by an 8×8 discrete cosine transform (DCT). Finally, the DCT coefficients are quantized and zigzag-scanned. A number of low-frequency coefficients of each color plane is selected beginning with the DC coefficient.

¹ The collections and topic annotations are publicly available under the Creative Commons Attribution-Noncommercial-Share Alike license, <http://creativecommons.org/licenses/by-nc-sa/3.0/> – please contact sebastian.stober@ovgu.de

Those coefficients form the descriptor (we obtain 3 different feature vectors – one for each color component – by concatenating the coefficients). We have chosen the recommended setting of 6, 3, 3 for the Y, Cr, Cb coefficients respectively.

2.3 Distance Computation

Facet Definition. Based on the features associated with the images, *facets* are defined that refer to different aspects of visual (dis-) similarity:

Definition 1. *Given a set of features F , let S be the space determined by the feature values for a set of images I . A facet f is defined by a facet distance measure δ_f on a subspace $S_f \subseteq S$ of the feature space, where δ_f satisfies the following conditions for any $x, y \in I$:*

- $\delta(x, y) \geq 0$ and $\delta(x, y) = 0$ if and only if $x = y$
- $\delta(x, y) = \delta(y, x)$ (symmetry)

Optionally, δ is a distance metric if it additionally obeys the triangle inequality for any $x, y, z \in I$:

- $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$ (triangle inequality)

Specifically, during the study, three facets were used – each one referring to a single one of the above MPEG-7 features in combination with the respective distance measure proposed in the MPEG-7 standard:

For comparing two CLDs ($\{Y^{(a)}, Cr^{(a)}, Cb^{(a)}\}$ and $\{Y^{(b)}, Cr^{(b)}, Cb^{(b)}\}$), the sum of the (weighted) euclidean color component distances is computed [7]:

$$\delta_{CLD}(a, b) = \sqrt{\sum_i w_{yi}(Y_i^{(a)} - Y_i^{(b)})^2} + \sqrt{\sum_k w_{rk}(Cr_k^{(a)} - Cr_k^{(b)})^2} + \sqrt{\sum_k w_{bk}(Cb_k^{(a)} - Cb_k^{(b)})^2} \tag{1}$$

where Y_i , represent the i th luminance coefficient and Cr_k, Cb_k the k th chrominance coefficient. The distances are weighted appropriately, with larger weights given to the lower frequency components [2]

The proposed distance for the SCDs of images a and b is the l_1 -norm on the coefficients $(c_1^{(a)}, \dots, c_n^{(a)})$ $(c_1^{(b)}, \dots, c_n^{(b)})$ in the Haar-transformed histogram domain:

$$\delta_{SCD}(a, b) = \sum_{i=1}^n |c_i^{(a)} - c_i^{(b)}| \tag{2}$$

For the EHDs, the l_1 -norm is used as well to compare the images a and b :

$$\delta_{EHD}(a, b) = \sum_{i=1}^{80} |h_i^{(a)} - h_i^{(b)}| + 5 \times \sum_{i=1}^5 |g_i^{(a)} - g_i^{(b)}| + \sum_{i=1}^{65} |s_i^{(a)} - s_i^{(b)}| \tag{3}$$

² Note that the CLD component weights are applied to compute the distance for the CLD facet and thus are part of the facet’s definition in contrast to the facet distance weights defined below that are used for the aggregation of different facets.

Here the h_i refer to the 5×16 histogram bin values, g_i to the 1×5 global-edge histogram bins (weighted by factor 5) and s_i to 13×5 semiglobal-edge histograms. All bin values are normalized.

Facet Distance Normalization. In order to be able to aggregate values from several facet distance measures, the following normalization is applied for all distance values $\delta_f(x, y)$ of a facet f :

$$\delta'_f(a, b) = \min \left\{ 1, \frac{\delta_f(a, b)}{\mu + \sigma} \right\} \quad (4)$$

where μ is the mean

$$\mu = \frac{1}{|\{(x, y) \in I^2\}|} \sum_{(x, y) \in I^2} \delta_f(x, y) \quad (5)$$

and σ is the standard deviation

$$\sigma = \sqrt{\frac{1}{|\{(x, y) \in I^2\}|} \sum_{(x, y) \in I^2} (\delta_f(x, y) - \mu)^2} \quad (6)$$

of all distance values with respect to δ_f . This truncates very high distance values and results in a value range of $[0, 1]$.

Facet Distance Aggregation. In order to compute the distance between images $a, b \in I$ w.r.t. to the facets f_1, \dots, f_l , the individual facet distances $\delta_{f_1}(a, b), \dots, \delta_{f_l}(a, b)$ need to be aggregated. Here, we use the weighted sum:

$$d(a, b) = \sum_{i=1}^l w_i \delta_{f_i}(a, b) \quad (7)$$

which is a very common weighted aggregation function that allows to control the importance of the facets f_1, \dots, f_l through their associated weights w_1, \dots, w_l . Per default, all weights are initialized as $\frac{1}{l}$, i.e. considering all facets equally important.

3 Adaptation Method

Changing the weights of the facet distance aggregation function described in the previous section allows to adapt the distance computation to a specific retrieval task. This can already be done manually – e.g., using the slider widgets (hidden on a collapsible panel) in the graphical user interface. However, it is often hard to do this explicitly. Several metric learning methods have already been proposed that aim to do this automatically: Generally, the first step is to gather preference information – either by analyzing information created by the user such as already labeled objects or manual classification hierarchies (e.g. documents in

a folder structure) [1] or alternatively by interpreting user actions such as rearrangement of objects in a visualization [12], changing cluster assignments [1], sorting a result list [11] or directly giving similarity judgments [2]. In the second step, the gathered preference information is turned into similarity constraints. These constraints are used finally to guide an optimization algorithm that aims to identify weights that violate as few constraints as possible. At this point, several possibilities exist: E.g., [12] describes a quadratic optimization approach that is deterministic and has the advantage of gradual and more stable weight changes and non-negative, bounded weights. However, it cannot deal with constraint violations. The approaches presented in [12] rely on gradient descent and ensemble perceptron learning instead. These methods allow constraint violation but may cause drastic weight changes. Further, they do not limit the value range of the weights which can however be achieved by modifications of the gradient descent update rule as proposed in [2].

In this paper, we interpret the problem of adapting the distance measure as a classification problem as proposed in [2]: The required preference information is deduced from the topic annotation already made by the user. We assume that images of the same topic are visually similar and that the respective visual features are covered appropriately by the facets introduced in the preceding section. For any pair of images a and b annotated with the same topic T , we can demand that they are more similar (or have a smaller distance) to each other than to any other image c not belonging to T :

$$d(a, b) < d(a, c) \quad \forall (a, b, c) | a, b \in T \wedge c \notin T \quad (8)$$

where d is the aggregated distance function defined in Equation 7. This can be rewritten as:

$$\sum_{i=1}^l w_i (\delta_{f_i}(a, c) - \delta_{f_i}(a, b)) = \sum_{i=1}^l w_i x_i > 0 \quad (9)$$

with $x_i = \delta_{f_i}(a, c) - \delta_{f_i}(a, b)$. Using the positive example $(x, +1)$ and the negative example $(-x, -1)$ to train a binary classifier, the weights w_1, \dots, w_l define the model (hyperplane) of the classification problem. This way, basically any binary classifier could be used here. We apply the linear support vector machine algorithm as provided by LIBLINEAR [3] that is faster and generates better results than the gradient descent approach used initially. However, with this approach, a valid value range for the weights cannot be enforced. Specifically, weights can become negative. We added artificial training examples that require positive weights (setting a single x_i to one at a time and the others to zero), but these constraints can still be violated.

4 Experiment 1: Assessing the Potential for Adaptation

The first question to be answered is: Given all knowledge about the topic assignments, how much can the performance be improved through adaptation of the facet weights? This gives us an estimation of the “ceiling” for the adaption

in simulation or application in real world. We consider the following three levels of adaptation:

1. **Topic-specific adaptation (Topic):** This is the most general form of adaptation. For each photo of the topic, a ranking of all photos in the collection is considered and constraints are derived that require images of the same topic to be ranked higher than others. The facet weights are then learned subject to the constraints from all rankings. This results in the highest number of constraints.
2. **Query-specific adaptation considering all photos from the topic (Query_All):** Here, facets weights are not adapted per topic but per query. To this end, only the single ranking for the query and the derived constraints are considered.
3. **Query-specific adaptation considering only the 5 nearest neighbors from the topic (Query_5NN):** This is a variation of the previous case with the difference that here only the 5 nearest neighbors from the topic are considered relevant instead of all images of the topic. This is the most specific adaptation with the lowest number of constraints.

In order to assess the retrieval performance, precision and recall were computed using each image that belongs to a topic as single query. [Figure 2](#) shows the averaged recall-precision curves per topic for the three adaptation levels and the baseline (no adaptation). Retrieval performance varies a lot from topic to topic: For topics with high diversity that have several sub-clusters of similar images (e.g., “Sagrada Familia”, “Lizards”, “Birds”), it tends to be much worse than for rather homogeneous topics with only few outliers (e.g., “Stone Hallway...”, “Owls”, “Ningaloo Reef”). Generally, an improvement over the baseline can be observed but it is mostly marginal. This indicates that either the facets are unsuitable to differentiate relevant from irrelevant images or the small number of facets does not provide enough degrees of freedom for the adaptation.

`Query_5NN` provides the best adaptation in the low recall area whereas at higher recall `Query_All` performs better. This is not surprising considering the above mentioned diversity and resulting sub-clusters within the topics. The more specific the adaptation the more likely it will consider only neighbors of the same sub-cluster as relevant and thus show superior performance for these images while this overfitting leads to a penalty when trying to find other images of the same topic (in the high recall range). `Topic` only leads to (small) improvements on rather homogeneous topics such as “Paintings” and “Aboriginal Art”. For the topics “Torii” and “Tibidabo”, its precision is close to zero and significantly below the baseline. These topics were perceived as especially difficult by the participants of the study because they are very divers and share visual similarity only at a higher level of detail. E.g., the vermilion color of the Torii is very remarkable but the respective objects often cover only a small portion of the image.

Summarizing, it can be concluded that this setting does not have much potential for adaptation it thus is unsuitable for a simulation of user-interaction.

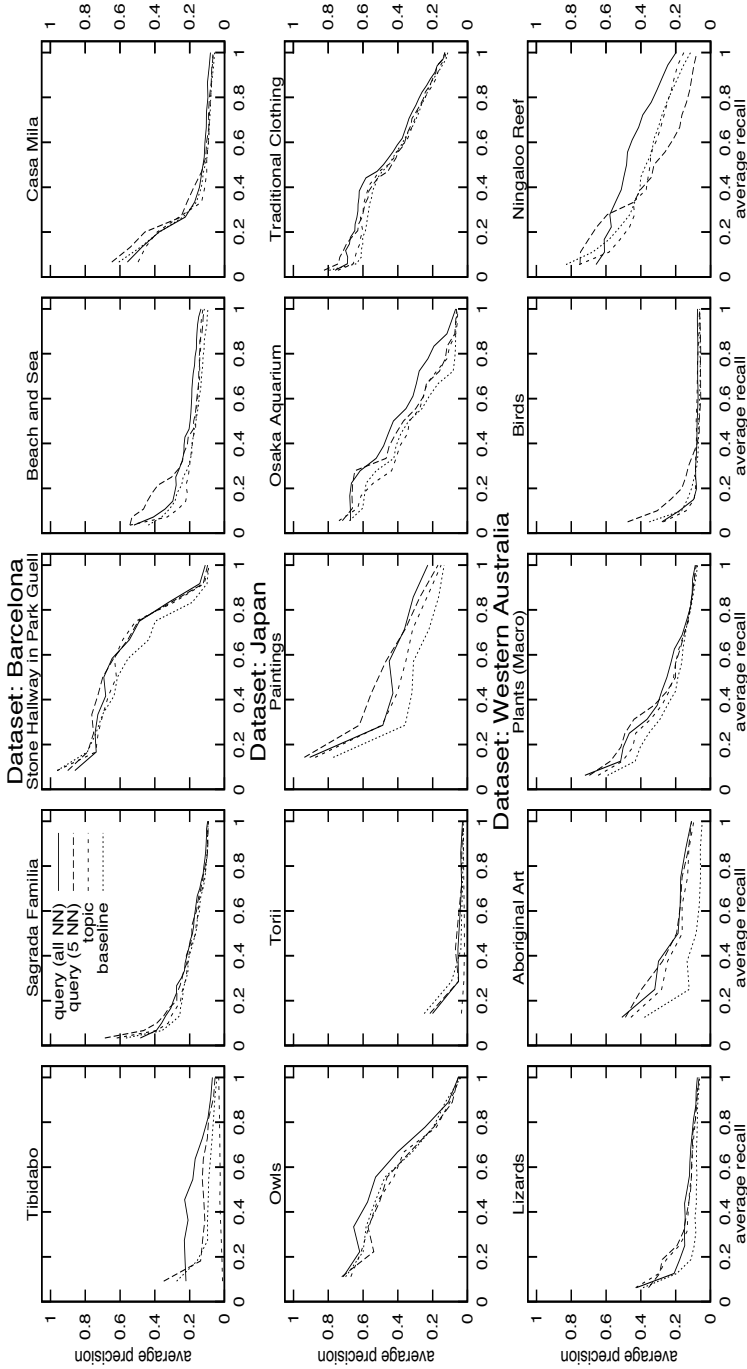


Fig. 2. Averaged recall-precision curves for each topic of the three evaluation datasets (rows) using 3 facets: ColorLayout, EdgeHistogram and ScalableColor. Each plot shows the performance of the initial weighting (baseline), the topic-specific adaptation and the query-specific adaptation (taking all or only the closest five images from the same topic).

The potential could be increased by adding more facets that cover different aspects of visual similarity which may help to differentiate the images of a topic from others. Alternatively, the information covered by the existing facets may in fact be already sufficient for differentiation but the comparison (i.e. the distance computation) takes place at a level that is too high to cover the inner-topic commonalities. For instance, the remarkable color of the Torii is captured by the SCD but in the current setting, we are only able to express that color in general is important for comparison but cannot stress this specific color. Similarly, it would make sense to emphasize vertical edges in the EHD for the Sagrada Família which is not possible either. In order to make such fine-grain adaptations possible, the respective sub-features currently hidden within the (high-level) facets need to be made visible for adaptation by becoming (low-level) facets themselves. This is covered by the next section.

5 Experiment 2: Extending the Number of Facets

As proposed in the previous section, this experiment investigates how the potential for adaptation can be increased by replacing a high-level facet by a set of low-level facets. Recall that a facet is defined by a set of features *and* a distance measure. Thus, in order to decompose a facet into sub-facets, it is not sufficient to just identify suitable subsets of the feature set (possibly splitting a feature further into sub-features). More importantly, it is also necessary to define appropriate distance measures that work on the feature subsets in a way that preserves the semantics of the original distance measure during aggregation by linear combination.

In the following, we consider the SCD as representative example for decomposing a facet. (For the EHD and CLD similar transformations can be done analogously using the histogram bins or the three coefficient vectors respectively as sub-features.) We choose the SCD because a finer differentiation in the color domain appears to be promising to increase performance on some of the difficult topics such as “Torii”. The decomposition of the SCD is very straightforward as its distance measure (Equation 2) is itself an (equally) weighted sum of per-coefficient distances. The SCD-facet can therefore simply be replaced by 64 SCD-coefficient-facets – each one considering a different coefficient and using the absolute value difference as facet distance measure. As a result, the adaptation algorithm has now 63 more degrees of freedom. Figure 3 shows the performance for running the experiment described in the previous section in this modified setting. The performance improvement is now clearly evident throughout all topic for all three adaptation levels. **Query_All** outperforms the others significantly, often achieving maximum precision in the low and middle recall range. **Query_5NN** does well on the first ranks but its precision rapidly declines afterwards which is a nice indicator for the overfitting that takes place here. **Topic** lies somewhere in between the baseline and **Query_All** – except for “Osaka Aquarium” where it has almost zero precision. This is somewhat surprising as this topic is rather homogeneous. Examining the topic weights learned by the

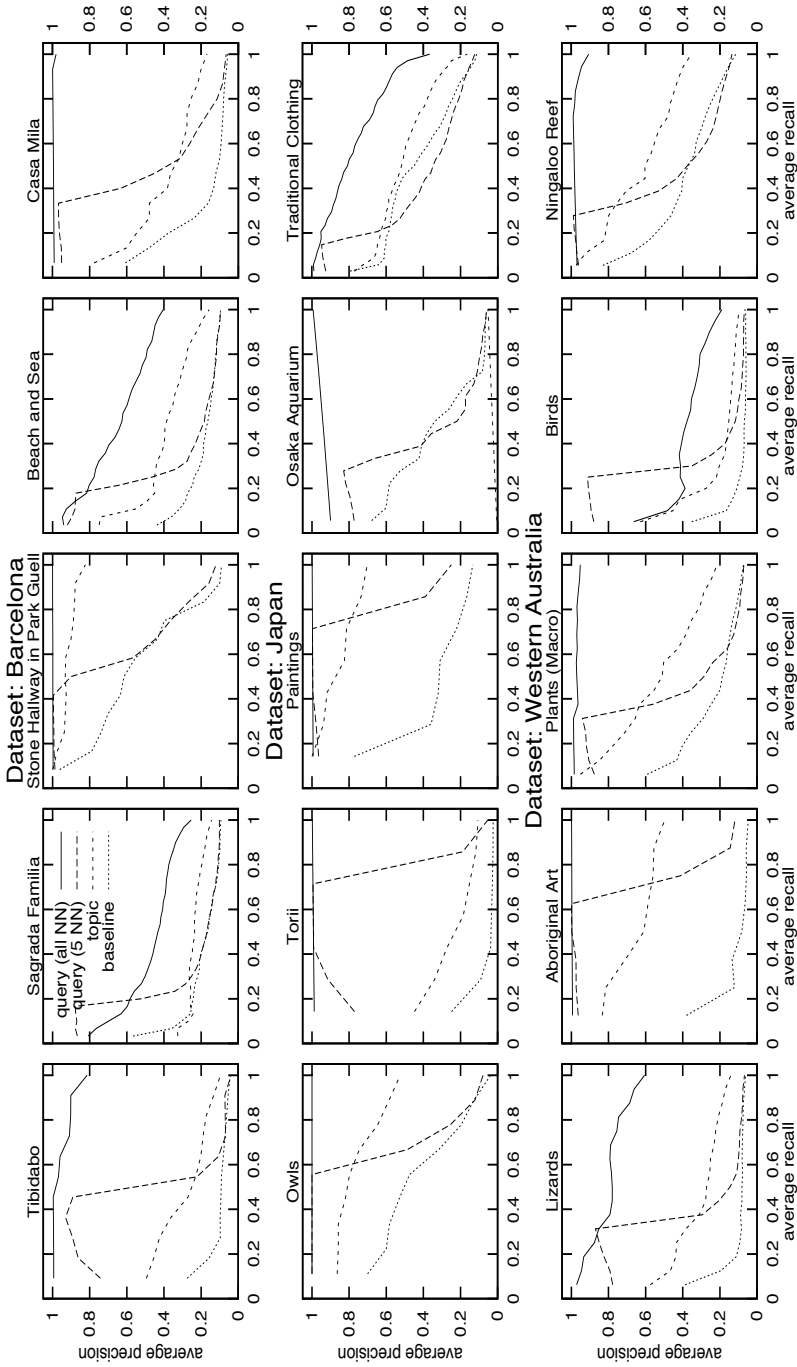


Fig. 3. Averaged recall-precision curves for each topic of the three evaluation datasets (rows) using 66 facets: ColorLayout (CLD), EdgeHistogram (EHD) and 64 bins of ScalableColor (SCD). Each plot shows the performance of the initial weighting (baseline), the topic-specific adaptation and the query-specific adaptation (taking all or only the closest five images from the same topic).

adaptation algorithm reveals that the weight are in the range $[-3.7, 3.6]$ with many negative values. This is caused by a known shortcoming of the adaptation method that cannot prohibit negative weights (see Section 3). Other weightings contain negative weights as well but are less extreme.

6 Experiment 3: Simulated User-Interaction

This experiment aims to simulate user-interaction as observed during the study. The question to be answered is whether an automatic weight adaptation could have helped the user in finding more relevant images for a topic through the secondary focus of the SpringLens. Figure 4 shows the outline of the simulation approach for a single session, i.e. finding five relevant images for a specific topic.

```

simulateSession(seed image seed, relevant images RELEVANT)
  initialize ANNOTATED  $\leftarrow$  {seed}
  repeat
    ANNOTATED  $\leftarrow$  ANNOTATED  $\cup$  findNextImage(ANNOTATED, RELEVANT)
    adapt weights
    evaluate
  until |ANNOTATED| = 5

findNextImage(annotated images ANNOTATED, relevant images RELEVANT)
  // try to find a relevant image in secondary focus
  for all query  $\in$  ANNOTATED do
    NN  $\leftarrow$  getKNearestNeighbors(query, 5)
    for all neighbor  $\in$  NN do
      if neighbor  $\in$  RELEVANT  $\setminus$  ANNOTATED then
        return neighbor
      end if
    end for
  end for
  // fallback: query with the newest annotated image
  newest  $\leftarrow$  newestIn(ANNOTATED)
  RANKING  $\leftarrow$  getKNearestNeighbors(newest,  $\infty$ )
  for all neighbor  $\in$  RANKING do
    if neighbor  $\in$  RELEVANT  $\setminus$  ANNOTATED then
      return neighbor
    end if
  end for

```

Fig. 4. Outline of the simulation algorithm

A first relevant image is required as *seed* for the simulation. After each additional relevant image, the weights are adapted considering the three levels of adaptivity introduced in Section 4 and evaluated afterwards. The simulated user's strategy to find the next relevant image relies on the secondary focus that contains the five nearest neighbors of the image in primary focus. (This is the same setting as in

the user study.) Directing the primary focus onto already annotated images, the users tries to find another relevant image in secondary focus. If this fails, he looks in the surrounding of the most recently annotated image (*newest*) – a region that is most likely not fully explored yet. This is simulated by going through the full similarity ranking of all images using *newest* as query and picking the first relevant image that has not been annotated yet. (As all topics contain more than five relevant images, this fallback strategy never fails.) To generate the ranking and for finding the five nearest neighbors, the **Query_All** weights are used that performed best in the previous experiments. (For the first query with the *seed*, no adaptation can be made because at least two annotated images are required.)

Figure 5 shows the performance after each iteration averaged over all *seed* images for each topics. W.r.t. the user’s retrieval goal – finding five images for each topic – two performance value are of interest: the precisions at rank 5 and the number of new relevant images in secondary focus. The *precisions at rank 5* refers to the portion of relevant images in secondary focus because it contains the five nearest neighbors. Looking only at this value, the adaptation increases performance significantly for **Query_All** and **Query_5NN** – both having identical values. For **Topic**, there is still an improvement in most cases. However, looking at the precision values, it has to be taken into account that with each iteration more relevant images are known to the user – and thus to the adaptation algorithm. It would be easy for an adaptation algorithm to overfit on this information by always returning simply the already annotated images as nearest neighbors. This way, a precision of 4.0 could be easily reached after four iterations. While such an adaptation could help to re-discover images of a topic, it is useless for the considered task of finding new relevant images. This issue is addressed by the *number of new images in secondary focus* performance measure. The values reveal that for the hard topics like “Torii” or “Tibidabo” the adaptation indeed leads to the above described overfitting and does not help much to find new images of the same topic. However, for about two thirds of all topics the adaptation turns out to be helpful as between 1 and 5 new relevant images can be found in the secondary focus. This value naturally decreases with each iteration as previously new images become annotated. Further, overfitting may also be involved to some extend but this cannot be measured.

7 Conclusion

We conducted three experiments to answer the question whether and how much automatic similarity adaptation can help users in an exploratory retrieval scenario where images are to be annotated with topic labels. As visual descriptors the EdgeHistogram, ScalableColor and ColorLayout Descriptors from the MPEG-7 specification were used. Similarity was adapted by changing the weights for several distance facets. The first experiment revealed that the initial setting – weighting three facet (one for each visual descriptor used) did not provide enough degrees of freedom for the adaptation approach. Decomposing the ScalableColorDescriptor into its bins introduced additional low-level facets and increased

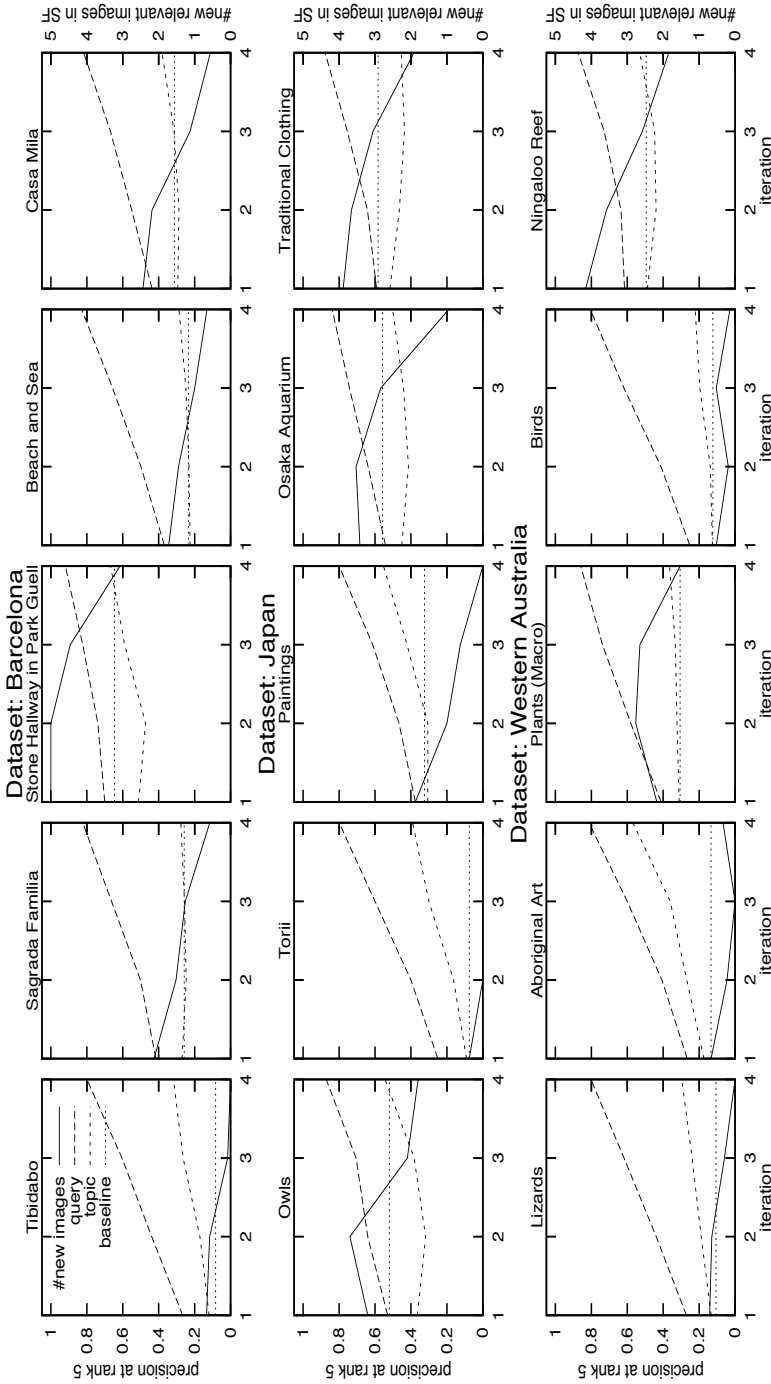


Fig. 5. Change of the performance over the course of the simulation (iterations) for each topic of the three evaluation datasets (rows) using 66 facets. Values are averaged over all possible seed images of the topic. Each plot shows the precision at rank 5 (left y-axis) for the initial weighting (baseline), the topic-specific adaptation and the query-specific adaptation. The solid line shows the average number of new relevant images in secondary focus (right y-axis) for the query-specific adaptation.

the potential for adaptation significantly as shown the second experiment. In the third experiment, user-interaction was simulated and the quality of the adaptation evaluated. It can be concluded the adaptation is useful in the considered retrieval scenario. The proposed decomposition approach is likely to work for other complex feature descriptors beyond those covered here. However, this still needs to be investigated more thoroughly.

Acknowledgments. This work was supported in part by the German National Merit Foundation and the German Research Foundation (DFG).

References

1. Bade, K., Nürnberger, A.: Creating a cluster hierarchy under constraints of a partially known hierarchy. In: Proc. of 8th SIAM Int. Conf. on Data Mining (2008)
2. Cheng, W., Hüllermeier, E.: Learning similarity functions from qualitative feedback. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 120–134. Springer, Heidelberg (2008)
3. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* 9, 1871–1874 (2008)
4. Germer, T., Götzelmann, T., Spindler, M., Strothotte, T.: Springlens: Distributed nonlinear magnifications. In: Eurographics 2006 - Short Papers (2006)
5. Kruskal, J., Wish, M.: *Multidimensional Scaling*. Sage, Thousand Oaks (1986)
6. Lux, M.: Caliph & Emir: MPEG-7 photo annotation and retrieval. In: Proc. of 17th ACM Int. Conf. on Multimedia (2009)
7. Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology* 11, 703–715 (1998)
8. Martinez, J., Koenen, R., Pereira, F.: MPEG-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia* 9(2), 78–87 (2002)
9. Stober, S., Hentschel, C., Nürnberger, A.: Evaluation of adaptive springlens – a multi-focus interface for exploring multimedia collections. In: Proc. of 6th Nordic Conference on Human-Computer Interaction, NordiCHI 2010 (2010)
10. Stober, S., Hentschel, C., Nürnberger, A.: Multi-facet exploration of image collections with an adaptive multi-focus zoomable interface. In: Proc. of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010) (2010)
11. Stober, S., Nürnberger, A.: Towards user-adaptive structuring and organization of music collections. In: Detyniecki, M., Leiner, U., Nürnberger, A. (eds.) AMR 2008. LNCS, vol. 5811, pp. 53–65. Springer, Heidelberg (2010)
12. Nürnberger, A., Stober, S.: User modelling for interactive user-adaptive collection structuring. In: Boujemaâ, N., Detyniecki, M., Nürnberger, A. (eds.) AMR 2007. LNCS, vol. 4918, pp. 95–108. Springer, Heidelberg (2008)
13. Stober, S., Nürnberger, A.: Musicgalaxy – an adaptive user-interface for exploratory music retrieval. In: Proc. of 7th Sound and Music Conference, SMC 2010 (2010)

Similarity Query Postprocessing by Ranking

Petra Budikova, Michal Batko, and Pavel Zezula

Faculty of Informatics, Masaryk University, Brno, Czech Republic
{petra.budikova,batko,zezula}@fi.muni.cz

Abstract. Current multimedia search technology is, especially in commercial applications, heavily based on text annotations. However, there are many applications such as image hosting web sites (e.g. Flickr or Picasa) where the text metadata are of poor quality in general. Searching such collections only by text gives usually rather unsatisfactory results. On the other hand, multimedia retrieval systems based purely on content can retrieve visually similar results but lag behind with the ability to grasp the semantics expressed by text annotations. In this paper, we propose various ranking techniques that can be transparently applied on any content-based retrieval system in order to improve the search results quality and user satisfaction. We demonstrate the usefulness of the approach on two large real-life datasets indexed by the MUFIN system. The improvement of the ranked results was evaluated by real users using an online survey.

Keywords: ranking, content-based retrieval, metric space.

1 Introduction

With the rapid growth of volume and diversity of the digital data, the need of efficient storage and retrieval methods is indisputable. The traditional databases are not suitable for many new complex data types, such as multimedia, DNA sequences, time series, etc. Therefore, new methods of data management have been intensively researched in recent years.

Multimedia retrieval systems usually use one of two general approaches. The first one applies existing text-search mechanisms to retrieve the data based on the descriptive annotations. Recently, this approach was enhanced using result ranking with respect to content-based similarity [4]. Of course, the quality of results depends on the quality of text metadata, which is often not very high (especially in large general-purpose collections, such as web image galleries).

The second approach retrieves data by content. Data objects are indexed and searched using features extracted from the data that describe their important characteristics. The query-by-example paradigm is usually used for searching, which enables a natural definition of a complex object query, e.g. an image. The so-called content-based searching has been developing rapidly in recent years and has already grown to the web dimension. However, it suffers from the well-known *semantic gap* problem, i.e. the discrepancy between the similarity as

computed using the descriptors and human understanding of similarity [14]. Existing solutions try to bridge the gap using semantics-learning mechanisms [8] or iterative query refinement using relevance feedback [18].

While the text-based searching can be very successful in some applications, its obvious drawback is that it cannot be used on data where the text metadata is of low quality or not available. Here, the content-based approach is the only possibility. To overcome the semantic gap problem, the search engine can be trained to recognize semantic categories. However, there are a number of scenarios when the semantics-learning cannot be employed, e.g. in case of large datasets with many ambiguous semantic categories, where the computer learning is infeasible. Therefore, we need a general solution for fast content-based searching in data with no (or poor) semantic information as a fall-back option for situations where more precise approaches cannot be used.

Even though the concept of similarity is subjective and context-dependent, the search engine usually employs a general measure of similarity to provide fast retrieval. As a result, the retrieved objects are similar to the query in some ways but may not be the most relevant according to the user. To demonstrate this, imagine a user who searches for images of red apple and, based on visual similarity, the system provides tomatoes and red balls in addition to red apples.

In this paper, we propose to overcome this problem by combining several views on the relative importance of target objects. This method has already been proved to be very successful in the text-based searching but has not yet been used in a large-scale content-based retrieval. To provide efficient searching, we first retrieve a candidate set of objects using a general similarity measure. In the next step, other measures are applied on the initial result to adjust the ranking of the objects so that the most relevant results are displayed to the user. This solution has a number of advantages:

- **Generality:** Since the existing technologies for content-based searching are typically based on the metric model, this approach enables to search efficiently in a wide scope of data domains, ranging from multimedia to DNA sequences. Even more general (non-metric) measures can be used in the ranking phase where only a small number of objects needs to be processed.
- **Query-by-example search:** In many data domains, it is often difficult to describe the required objects by text or other attributes – “an image is worth a thousand words”. The content-based search enables an example object to be used to define the query.
- **Multi-modal searching:** The ranking concept enables to combine more similarity measures. In particular, it can easily employ similarity measures that are computationally expensive as well as to use information that is not rich enough to provide a full-fledged result on its own.
- **Flexibility:** There are several sources of information that can be exploited in the search process. The search engine has the knowledge of the data properties, can compute distances between pairs of objects, and use statistical information about the collection. In addition, the system can interact with the user or other systems to adjust or re-evaluate the ranking procedure.

The rest of the paper is organized as follows. In Section 2, we discuss the most relevant related work. We briefly introduce the content searching based on metric space similarity in Section 3. Next, we formalize our concept of two-phase searching in Section 4 and propose a basic classification of ranking methods. User-satisfaction experiments with several ranking functions are described and evaluated in Section 5.

2 Related Work

The concept of query result post-processing and ranking has been employed in a number of search applications and strategies, both in text-based and similarity-based retrieval. Most of the research has been done in image and video searching, which is attractive for many users. In the text-based approaches, ranking is often used to prioritize objects from the result that have similar visual content as the query object. In content-based strategies, various post-processing methods try to bridge the semantic gap and identify the most relevant objects.

The text-based search in images has been provided by many web search engines for years. Recently, some of the major search engines (Google¹, Bing²) launched a new type of searching based on visual similarity of images. Both solutions exploit visual ranking of search results acquired by text retrieval. The Google approach⁴ employs local image descriptors to measure the visual similarity of images. The famous *PageRank* algorithm idea has been adapted to *VisualRank*, which is used to propagate the similarity relationships in the result. Since the complete evaluation of the ranking algorithm is expensive, the results of visual search are precomputed for the more popular queries.

The Microsoft solution¹⁶ is based on a similar concept, this time using both local and global visual image descriptors to rank objects retrieved by the initial text search. To obtain more precise results, the descriptors receive weights that express their importance for that particular image set. Again, the image set is modeled by a visual-similarity graph and the similarity information is propagated to identify the most important nodes.

An interesting extension to these methods has been proposed in⁹. The authors argue that the results of the visual ranking are often not satisfactory, which is caused by the fact that the initial text-based search result is not good enough to allow detecting important patterns for ranking. To overcome this, they propose to combine results from multiple web search engines and provide the *CrowdRanking* algorithm which identifies important visual features and ranks the results.

The text information associated with multimedia objects is often in the form of *tags*, i.e. keywords provided by users. Tagging is popular especially in social media repositories such as Flickr³ and can be exploited in search processes. The authors of⁶ investigate ways of differentiating between content-related and

¹ <http://images.google.com/>

² <http://www.bing.com/images>

³ <http://www.flickr.com>

content-unrelated tags by the means of WordNet relationships between semantic concepts. Another study [7] explores the ways of determining ranking of tags according to their relevance.

One of the weaknesses of text-based search is the ambiguity of search terms. An active ranking strategy was proposed in [15] where a small set of images that represent different concepts is chosen from the result of the text-based search and displayed to the user for evaluation of relevance. The user input is used to disambiguate the search.

In case of results obtained by content-based searching, the post-processing methods try to filter out less interesting objects from the result, usually by means of result clustering. Two quality aspects that are mostly addressed are the presence of too many near-duplicates in the result set and the occurrence of objects that are not relevant from the user's point of view. For the first problem, a definition of a new *distinct nearest neighbors* query is provided in [13]. Such query only returns distinct objects, i.e. objects with mutual distances greater than some predefined constant denoted as the separation distance. To eliminate the less relevant objects, several methods have been proposed that try to analyze the relationships between the objects in the result set and identify the ones that are most important in some sense, e.g. they are most similar to the rest of the result. In [12], four methods of result ranking using clusters are presented, e.g. by penalizing objects in clusters other than the query object's cluster. The authors of [5] propose to use dynamic clustering, where the distance function for clustering is chosen with respect to the importance of individual features for the given query.

3 Content-Based Searching Using Metric Spaces

In our approach, the similarity is modeled by using a generic metric space abstraction. The image visual descriptors that are usually used in the field of image retrieval, e.g. the global descriptors defined in the MPEG-7 [10], in fact satisfy properties of the metric spaces and thus can be used in metric-based indexing engines. In order to work with very large collections of data, we employ the scalable indexing infrastructure of the Multi-Feature Indexing Network [11]. It is a versatile and highly modular similarity framework built on top of MESSIF library [2] which provides indexing layers as well as user and programming interfaces. Since the system works with any metric data, it allows us to work with a wide variety of data types including images.

3.1 Metric Space Approach

The metric space [17] is considered to be the most general data model for similarity search which can still be indexed and searched efficiently. The model treats its data as unstructured objects together with a function which measures proximity of object pairs. Formally, the *metric space* \mathcal{M} is a pair $\mathcal{M} = (\mathcal{D}, d)$, where \mathcal{D} is the *domain* of objects and d is a total *distance function* $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ satisfying

the following postulates for all objects $x, y, z \in \mathcal{D}$: the reflexivity $d(x, x) = 0$, the strict positiveness $d(x, y) > 0$, the symmetry $d(x, y) = d(y, x)$, and the triangle inequality $d(x, y) \leq d(x, z) + d(z, y)$.

The semantics of this concept is: The smaller the distance between two objects, the more similar they are. The metric space is typically searched by queries which follow the query-by-example paradigm. A query is formed by an *object* $q \in \mathcal{D}$ and some *constraint* on the data to be retrieved from the indexed dataset $X \subseteq \mathcal{D}$. There are two basic types of these queries: (1) the *range query* $R(q, r)$, which retrieves all objects $o \in X$ within the range r from q , and (2) the *nearest neighbors query* $kNN(q, k)$, which returns the k objects from X with the smallest distances to q .

3.2 Multi-Feature Indexing Network

The Multi-Feature Indexing Network (MUFIN) [11] is a general purpose search engine that exploits results of more than ten years of research of the metric-based techniques. It allows to plug-in different metric indexing techniques and bind them together by various programming interfaces into a coherent system. A strong emphasis is put on scalability, so the system can process data collections of really big volumes. Moreover, the system provides easy-to-use user interfaces that can be used to offer the searching capabilities to any user. The system is also able to gather various statistics and thus can be easily adjusted to perform user satisfaction surveys and testing new searching paradigms.

In this paper, we use two instances of the MUFIN system for evaluating the results of content-based similarity queries. The capabilities of MUFIN allowed us to retrieve the results very fast even for large collections of data and its interfaces made it possible to plug-in the proposed ranking algorithms seamlessly into the web user interfaces.

4 Ranking

Ranking is often considered an integral part of the search process – search engines retrieve ranked results. However, we argue that it is more convenient to treat searching as a two-phase process, distinguishing between the initial search phase, which retrieves suitable candidates, and the ranking phase. The crucial difference between these phases is that in the first phase, the whole dataset is searched while only a small subset is processed in the second phase.

Let us now formalize the two search phases as functions over the data space $\mathcal{M} = (\mathcal{D}, d)$. The initial search $F_{initial}$ may be performed by any standard metric search query operation returning a set of k objects relevant to the given query object q , e.g. the k -nearest neighbor search (kNN) or the range search. In our framework, we choose the kNN search as the most convenient, since users do not need any prior knowledge about the distances in the dataset.

$$F_{initial}(q) = R \subseteq \mathcal{D}, |R| = k$$

In the ranking phase, a function $F_{rank} : \mathcal{D} \mapsto \mathbb{N}$ is applied on the result of the initial search $F_{initial}$ to establish a new rank of each object. In fact, the ranking function depends on the *context* in which it is evaluated and its computation may contain additional context-derived parameters. To increase the readability we relax the strictness of the function definition by including the context parameters in $RANK_{type}$ function as needed. We will discuss the possible context parameters later.

$$F_{rank}(o) = RANK_{type}(o, context) = i,$$

i is the rank of the object $o \in F_{initial}$ in the given context

The ranking function F_{rank} must satisfy the following *unambiguity condition*:

$$\forall o_1, o_2 \in F_{initial}(q) : (F_{rank}(o_1) = F_{rank}(o_2)) \Rightarrow (o_1 = o_2).$$

Even though the user is interested in the first k objects with k typically ranging from 10 to 100, the initial search should provide significantly more objects in order to allow the ranking to show interesting new data. Note that the larger the initial result set is, the higher the chances of having more relevant objects are. On the other hand, if the initial result is too large, the post-processing step might be too costly. Therefore, the choice of the initial result size k' needs to balance the following three factors: the costs of the initial search for k' best objects, the cost of ranking the k' objects, and the probability that there are at least k relevant objects in the initial result of size k' .

In the following sections, we present several different types of ranking functions that are orthogonal to the content-based similarity. Thus, the visual similarity of the image is supplemented by its semantic content expressed by textual annotation. We split the ranking functions into two categories – functions that can automatically rank the initial results based on the retrieved data and user-defined ranking where users actively participate in the process of defining the ranking function.

4.1 Automatic Ranking

As automatic we denote ranking methods that compute the result ranking using only the query context information, i.e. the query definition and the statistical properties of the initial result R . When the initial set is retrieved by a visual content, a successful ranking needs to exploit additional information available for data objects that was not used in the initial content-based search, e.g. keywords, location, searching object popularity, number of purchases of the object, etc. A more sophisticated ranking can try to identify and exploit some patterns in the properties of objects in the initial result, e.g. the most important keywords, or visual features in case of images. Finally, the ranking phase may also include another type of content-based similarity search. Naturally, several ranking functions can be combined to provide the final order of objects.

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

Keyword Ranking. Inversely to the search model applied by the common web search engines that combine text-based retrieval and visual ranking, we propose to rank the content-based search result with respect to keywords of the query image. We measure the similarity between two sets of keywords by the Jaccard coefficient (see [17] for a formal definition of the Jaccard similarity).

$$\begin{aligned} RANK_{queryObjectKeywords}(o, R, q) = i \in \mathbb{N}, i = |X|, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(q.keywords, x.keywords) \\ < d_{Jaccard}(q.keywords, o.keywords)) \end{aligned}$$

This ranking method is intended for data with rich and reliable annotations. In order to broaden the ranking range, we apply stemming and use WordNet to retrieve the keywords from semantic relationships as suggested in [6]. Using the WordNet, we also remove all words that are not nouns, verbs or adjectives.

Word Cloud Ranking. For data with sparse and erroneous text metadata, the keyword ranking is not applicable. In this case, we propose to exploit the keywords of all objects in the initial result. The keywords are first cleaned and broadened by WordNet as anticipated above. Then we compute the frequencies of the keywords from all the objects in the initial result. We call the resulting set of keywords with their frequencies the *word cloud*. Finally, the ranking employs the most n frequent words from the cloud (denoted as $R.wordCloud.top(n)$) as the query object words in the text-similarity evaluation. Please note that the object keywords $o.keywords$ in the following definition are the keywords of the respective object cleaned by the WordNet as described above.

$$\begin{aligned} RANK_{wordCloud}(o, R, q, n) = i \in \mathbb{N}, i = |X|, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(R.wordCloud.top(n), x.keywords) \\ < d_{Jaccard}(R.wordCloud.top(n), o.keywords)) \end{aligned}$$

Combined Visual and Text Ranking. In the previous methods, we have only used the textual (keyword) information for the ranking, ignoring the initial ranking of the visual (content-based) search. However, since the initial result is retrieved using the kNN query which provides the ranking of its own (the metric distance to the query object q), it may also be useful to add it into the final ranking. Therefore, we enrich the $RANK_{queryObjectKeywords}$ method by summing with the distance of the respective object from the visual space. Note that since the Jaccard measure gives values between zero and one, we need the visual distance to be normalized so that both of the two summed distances influence the ranking accordingly. Thus, we multiply the visual distance by a normalization factor f (e.g. the maximal distance in the dataset).

$$\begin{aligned} RANK_{queryObjKwAndVisual}(o, R, q) = i \in \mathbb{N}, i = |X|, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(q.keywords, x.keywords) + f \cdot d(q, x) \\ < d_{Jaccard}(q.keywords, o.keywords) + f \cdot d(q, o)) \end{aligned}$$

Adjusting the factor f can also be used to strengthen or diminish the impact of the visual descriptors on the ranking. Moreover, the $RANK_{wordCloud}$ can be

modified in a similar fashion resulting in the $RANK_{wordCloudAndVisual}$ function that combines the results of the word cloud ranking with the visual distances.

Adaptive Keyword/Cloud Ranking. For datasets where some objects are poorly annotated or there is no annotation at all but some objects have a good metadata, it can be beneficial to adaptively choose a ranking method. Therefore, we propose the following heuristic that combines the previous raking methods. Given the query object's keywords and the word cloud of the initial result, we prepare the set of adaptive keywords A as follows. First, all the cleaned keywords of the query object are inserted. If there are less than c of these, the most frequent cloud words are added. However, the cloud words must exhibit some minimal frequency t to be considered relevant. Note that the WordNet cleaning and enrichment as defined above is used. The final ranking is computed as a combination of the text ranking defined by the described keyword set and the initial visual ranking.

$$\begin{aligned}
 RANK_{adaptive}(o, R, q, c, t) &= i \in \mathbb{N}, i = |X|, X \subseteq R, \\
 A &= q.keywords \cup R.wordCloud.top(c - |q.keywords|, t), \forall x \in X : \\
 &\quad (d_{Jaccard}(A, x.keywords) + d(q, x)) \\
 &\quad < d_{Jaccard}(A, o.keywords) + d(q, o)
 \end{aligned}$$

4.2 User-Defined Ranking

As we have discussed in the introduction, the understanding of similarity is subjective and varies in different conditions. Therefore, it is not always possible to obtain the optimal result automatically and the user needs to cooperate with the system. In this case, the system displays the results of the initial search and requires additional user input for the ranking phase. A new query object, a measure of the relevance of the initial result, or a specification of relevant values for associated object metadata are a few examples of possible user input for the ranking phase.

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

Relevance Feedback Ranking. In some search systems, users can provide a feedback on the relevance of results and ask for a refined result. To provide this, the system uses the relevance information to modify the query object or the similarity measure (see [18] for more details). This may be repeated in several iterations which finally produce a better result but may take a considerable amount of time, as a new query needs to be evaluated in each iteration. Therefore, we propose to implement the relevance feedback as the ranking function. Users choose relevant objects from the initial result and the ranking function defines the final rank as a function on the content-based similarity to each of the objects marked as relevant.

$$\begin{aligned} RANK_{relevanceFeedback}(o, R, d_{agg}, [q_1, \dots, q_n]) = \\ i \in \mathbb{N}, i = |X|, X \subseteq R, \forall x \in X : \\ d_{agg}(d(q_1, x), \dots, d(q_n, x)) < d_{agg}(d(q_1, o), \dots, d(q_n, o)) \end{aligned}$$

Any monotonic function can be used as the aggregation function d_{agg} , for instance SUM, MIN or MAX.

User-Defined Keyword Ranking. Keywords may provide a strong ranking tool but automatic approaches may not always guess the optimal set of words. This method allows users to define the relevant keywords themselves.

$$\begin{aligned} RANK_{selectedKeywords}(o, R, keywordSet) = \\ i \in \mathbb{N}, i = |X|, X \subseteq R, \forall x \in X : \\ (d_{Jaccard}(keywordSet, x.keywords) \\ < d_{Jaccard}(keywordSet, o.keywords)) \end{aligned}$$

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

5 Experiments

To evaluate the quality of all the ranking functions defined in Section 4, we have organized several user-satisfaction surveys. We have provided a simple web interface where the participants were shown the initial and the ranked result sets and then they had to mark the relevant objects. In the two cases of user-defined ranking, the participants were first asked to choose the relevant objects/words from the initial result and then they evaluated the new ranking. About 40 users of different age, sex and computer skills participated in the experiments.

For the experiments, we used two different datasets. Dataset 1, which comes from a commercial microstock site, contains high-quality images with rich and systematic annotations. This dataset contains 8.3 million images and the content-based similarity is defined as a combination of *color layout*, *scalable color*, *region shape* and *edge histogram* MPEG-7 descriptors [10]. Each image is annotated by about 25 keywords on average. Dataset 2 contains images from the Flickr web site and exhibits worse quality of images and sparse and erroneous keywords. This dataset is formed by 100 million images each of which is represented by five MPEG-7 descriptors (see [3] for more information). The effectiveness of the visual search in Dataset 2 using the MUFIN system was published in [1]. The results indicate that even though the effectiveness is satisfactory, there is still space for improvement.

In each set of experiments, we used 50 randomly chosen query objects. For an easy visualization of several result sets on a screen, we only used a result set with 10 objects. In the initial nearest neighbor search we always retrieved 200 objects, which were conveyed to the ranking function.

We express the user-perceived quality of each result as the ratio of the number r of objects marked as relevant to the number t of all displayed objects from

the result. We denote this measure as the *result quality* throughout this section. Note that this measure is not the same as the well known *precision* metric, since in our case, there is no ground truth to compare with. In fact, the understanding of similarity is highly subjective and therefore each user may have his/her own ground truth. Consequently, 100 % quality may not be reachable using this measure if there are less than t relevant objects in the dataset. Unfortunately, there is no feasible way of determining the individual ground truth sets.

5.1 User-Defined Ranking

In this section, we summarize the results of the user-satisfaction with the two previously defined user-interactive ranking functions. Apart from evaluating their performance, we also used the experiments to find out about the usefulness of the ranking in principle – we asked the users for their opinion whether they want to try ranking for each result. About 50 % of results over Dataset 2 (the worse one) and 72 % of results over Dataset 1 were considered worth trying; the rest of the result sets was either perceived as already too good (17 % for Dataset 1) or too bad (33 % for Dataset 2). In case of Dataset 2, we remark that the low quality of results as perceived by users is caused by the low quality of some of the randomly picked query images rather than bad performance of the initial searching.

Relevance-Feedback Ranking. We ran a set of experiments on each of the two datasets to test the $RANK_{relevanceFeedback}$. For the Dataset 1 (which is smaller), we also evaluated a *multi-object query* in order to compare the ranking results with the precise evaluation. The multi-object query $mkNN(q_1, q_2, \dots, q_n, k)$ retrieves k objects that are most similar to a set of given query objects, i.e. objects that have the k lowest sums of distances to each query object $d(q_1, o) + d(q_2, o) + \dots + d(q_n, o)$. Note that a precise answer for user supplied feedback objects can be retrieved by this query.

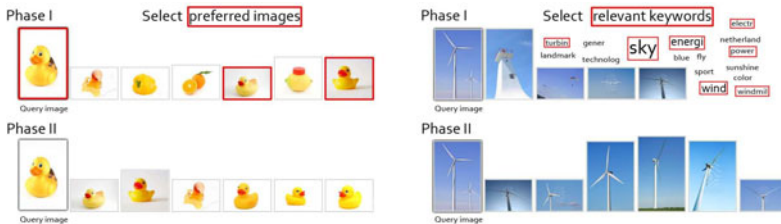


Fig. 1. User-defined ranking: relevance-feedback ranking (left), user-defined text ranking (right)

In the experiments, users were asked to choose any number of relevant objects from the displayed top-10 images from the initial result. Figure 1 shows an initial result set (Phase I) with the user-selected preferences marked by red border

and the final result set after the ranking was applied (Phase II). The actual aggregation function used in the ranking process was SUM.

In the following table, we compare the quality of results obtained by initial searching, ranking, and multi-object query evaluation.

	Dataset 1	Dataset 2
	result quality	result quality
$R_{initial}$	39.5 %	35.3 %
$RANK_{relevanceFeedback}$	59.2 %	48.0 %
<i>Multi-object query</i>	60.2 %	—

We can observe that the number of relevant objects in the initial result (i.e. ranked by the content-based similarity to the single query object) is increased significantly in both the experiments. Moreover, the ranking produces results of nearly the same quality as the full evaluation of the respective multi-object query, which finds the images most similar to all the query objects selected by the user precisely from the whole dataset. This confirms our assumption that there are enough good objects in the initial result.

Text Ranking. The ranking based on users' choice of keywords was evaluated only for the Dataset 1. Participants of the experiment were shown the initial result and a set of keywords, which comprised all keywords of the query object combined with the 50 most frequent keywords from the word cloud – we have not shown all the keywords to keep the list accessible. Different font sizes were used for the display of the keywords to emphasize the most frequent ones, as depicted in Figure 10 (right). Users were asked to choose any number of relevant keywords and evaluated the ranked result. The following table summarizes their satisfaction.

	Result quality
Initial result	34.1 %
Ranked result	48.6 %

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

Consistency of User's Preferences. We can make some observations on the behavior of users during the search process, which may be useful for further improvements of the automatic ranking methods. Although the users were not given any advice on how to decide the relevance, their evaluation of (ir)relevance

of given object in a particular result was very consistent – on average, more than 80% of users agreed on a relevance of a given object. In the phase of selecting words or images for ranking, the same object was chosen as preferred by 60-70% of users on average. This implies that some kind of a general truth exists that is favorable in most situations. It is therefore realistic and reasonable to develop automatic methods that try to find the relevant objects by the analysis of human preferences.

Selection of Initial Result Size. We have also focused on the changes of effectiveness when the size of the initial set k' is increased. In particular, Figure 2 shows the percentage of the relevant objects in the results set as specified by the users when the k' was varied from 10 to 200 on Dataset 1. The same user-defined ranking and $k = 10$ final results as in the previous experiment were used.

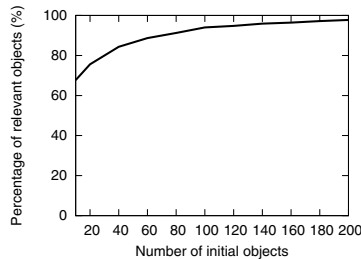


Fig. 2. Influence of the initial result set size on the number of relevant objects

As expected, we can observe that the effectiveness of the ranking increases with the size of the initial set. The improvement is increasing quickly as first, but as the size of the initial set contains more objects fewer relevant objects appear in the set reaching nearly 97% of all relevant objects at the size 200.

5.2 Automatic Ranking

Another set of experiments was designed to test the performance of the proposed automatic ranking methods over the two datasets with different characteristics. In this case, participants of the experiments were shown several sets of results on one page and asked to mark the relevant ones. Figure 3 shows a part of one such screen.

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

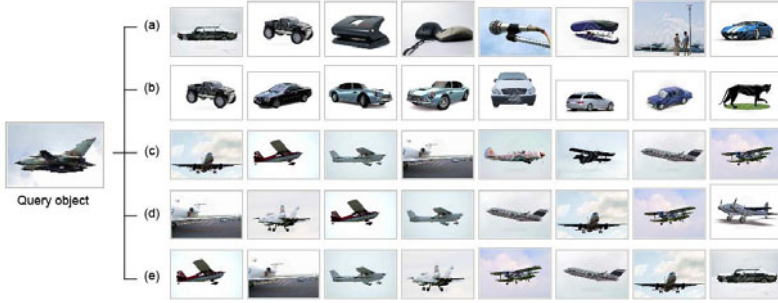


Fig. 3. Automatic ranking: a) $R_{initial}$, b) $RANK_{wordCloud}(o, R, q, 5)$, c) $RANK_{wordCloud}(o, R, q, 10)$, d) $RANK_{queryObjectKeywords}(o, R, q)$, e) $RANK_{queryObjKwAndVisual}(o, R, q)$

	Dataset 1 result quality	Dataset 2 result quality
$R_{initial}$	36.2 %	23.5 %
$RANK_{wordCloud}(o, R, q, 5)$	33.2 %	25.4 %
$RANK_{wordCloudAndVisual}(o, R, q, 5)$	41.3 %	32.5 %
$RANK_{wordCloud}(o, R, q, 10)$	35.1 %	24.9 %
$RANK_{wordCloudAndVisual}(o, R, q, 10)$	42.0 %	33.7 %
$RANK_{queryObjectKeywords}(o, R, q)$	55.4 %	41.1 %
$RANK_{queryObjKwAndVisual}(o, R, q)$	56.8 %	43.0 %
$RANK_{adaptive}(o, R, q, 10, 10)$	56.8 %	45.4 %

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

Let us recall here that the quality of results is upper-bounded by the quality of the data and in many cases it is not possible to obtain 100 % quality. However, for any number of relevant objects in the dataset, a good retrieval system should rank them on the top positions. Therefore it is reasonable to compare search results with respect to the rank of relevant objects. A possible metric used for this purpose is the *Spearman footrule* [17], which requires the ground truth. As we do not have this, we proposed a different measure of rank quality, which we call *sparseness*. This metric is defined as the average number of irrelevant objects between two adjacent relevant ones. In an optimal search, the sparseness of a result is 0. The following table shows this measure evaluated for the initial result and the best of our ranking methods.

	Dataset 1	Dataset 2
	result sparseness	result sparseness
$R_{initial}$	1.43	1.29
$RANK_{adaptive}(o, R, q, 10, 10, 10)$	0.63	0.67

5.3 Processing Time

As one of our objectives is effective and efficient processing of large datasets, we also need to discuss the relationships between obtained quality and computation costs. The initial searching exploits a scalable and efficient metric search infrastructure (see Section 3 for more details) which provides retrieval with nearly constant costs. The average response time of the initial search in this implementation is 500 ms. The ranking phase costs depend on the number of processed objects. The average time needed for post-processing of a dataset with 200 objects is about 30 ms. Let us recall that the post-processing provides results of a quality comparable to the results of the multi-object query, which guarantees precise results (see Section 5.1). However, the costs of a precise evaluation of the multi-object query is much higher, ranging from seconds to tens of seconds.

6 Conclusion

In this paper, we have focused on improving the quality of results retrieved by content-based search engines via ranking. In our scenario, first an initial result set is retrieved using a standard search engine. Then, a ranking function is applied on the results to push the more relevant objects to the top of the rank list using an orthogonal similarity measure. This approach has several benefits – it can be applied to any search engine, there are no restrictions on the ranking function, and it allows to combine orthogonal views on the returned objects without computationally expensive combination techniques.

In particular, we have retrieved images by visual content and then ranked the result using text annotations. We have compared 7 different automatic ranking methods that worked on the images’ keywords and 2 user-defined ranking methods where the feedback from the user was gathered. Since the similarity of images is subjective, we have measured the quality improvements by several user surveys with about 40 users of different age, sex and computer skills. Our experiments show that the ranking improved the satisfaction of users significantly – it has nearly doubled the quality of the results. We have also shown that even though the query result set still contains some irrelevant objects, the most relevant ones were pushed to the top.

The performance of the ranking methods depends heavily on the relevance of data objects in the initial result set. In the experiments, we have verified our assumption that there is a significant amount of relevant objects in the result of a general content-based search that are scattered among other objects and thus do not appear on the first result page. When several hundred top-ranking objects are submitted to the ranking method, the final result is comparable to the result of a much more expensive query processing over the whole dataset.

Acknowledgments. This work has been supported by the national research projects GACR 201/08/P507, GACR 201/09/0683 and by Brno PhD Talent Financial Aid. The hardware infrastructure was provided by the METACentrum under the research intent MSM6383917201.

References

1. Batko, M., Kohoutkova, P., Novak, D.: CoPhIR image collection under the microscope. In: Proceedings of SISAP 2009, pp. 47–54 (2009)
2. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: Thanos, C., Borri, F., Candela, L. (eds.) Digital Libraries: Research and Development. LNCS, vol. 4877, pp. 1–10. Springer, Heidelberg (2007)
3. Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F.: CoPhIR: a test collection for content-based image retrieval. CoRR abs/0905.4627 (2009)
4. Jing, Y., Baluja, S.: VisualRank: Applying PageRank to large-scale image search. *IEEE Trans. on Pattern Analysis and Machine Intell.* 30(11), 1877–1890 (2008)
5. van Leuken, R.H., Garcia, L., Olivares, X., van Zwol, R.: Visual diversification of image search results. In: Proc. of the 18th International Conference on World Wide Web, pp. 341–350. ACM, New York (2009)
6. Liu, D., Hua, X.S., Wang, M., Zhang, H.J.: Retagging social images based on visual and semantic consistency. In: Proceedings of WWW 2010, pp. 1149–1150. ACM, New York (2010)
7. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: Proceedings of WWW 2009, pp. 351–360. ACM, New York (2009)
8. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40(1), 262–282 (2007)
9. Liu, Y., Mei, T., Hua, X.S.: Crowdreranking: exploring multiple search engines for visual search reranking. In: Proceedings of SIGIR 2009, pp. 500–507. ACM, New York (2009)
10. MPEG-7: Multimedia content description interfaces. Part 3: Visual. ISO/IEC 15938-3:2002 (2002)
11. Novak, D., Batko, M., Zezula, P.: Generic similarity search engine demonstrated by an image retrieval application. In: Proceedings of SIGIR 2009, p. 840 (2009)
12. Park, G., Baek, Y., Lee, H.K.: Web image retrieval using majority-based ranking approach. *Multimedia Tools and Applications* 31(2), 195–219 (2006)
13. Skopal, T., Dohnal, V., Batko, M., Zezula, P.: Distinct nearest neighbors queries for similarity search in very large multimedia databases. In: Proceedings of WIDM 2009, pp. 11–14. ACM, New York (2009)
14. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intell.* 22(12), 1349–1380 (2000)
15. Tian, X., Tao, D., Hua, X.S., Wu, X.: Active reranking for web image search. *Trans. Img. Proc.* 19(3), 805–820 (2010)
16. Wang, L., Yang, L., Tian, X.: Query aware visual similarity propagation for image search reranking. In: Proceedings of MM 2009, pp. 725–728. ACM, New York (2009)
17. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. In: *Advances in Database Systems*, vol. 32. Springer, Heidelberg (2006)
18. Zhou, X.S., Huang, T.S.: Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems* 8(6), 536–544 (2003)

Proximity-Based Order-Respecting Intersection for Searching in Image Databases

Tomas Homola, Vlastislav Dohnal, and Pavel Zezula

Faculty of Informatics
Masaryk University
Botanicka 68a, 602 00 Brno
Czech Republic

Abstract. As the volume of non-textual data, such images and other multimedia data, available on Internet is increasing. The issue of identifying data items based on query containment rather than query equality is becoming more and more important. In this paper, we propose a solution to this problem. We assume local descriptors are extracted from data items, so the aforementioned problem reduces to finding data items that share as many as possible local descriptors with the query. In particular, we define a new ε -intersection for this purpose. Local descriptors usually contain the location of the descriptors, so the proposed solution takes them into account to increase effectiveness of searching. We evaluate the ε -intersection on two real-life image collections using SIFT and SURF local descriptors from both effectiveness and efficiency points of view. Moreover, we study the influence of individual parameters of the ε -intersection to query results.

Keywords: proximity-based order-respecting intersection, sub-image search, image database, experimental trials.

1 Introduction

The complexity of search in current business intelligence systems, academic research, or even the home audiovisual databases, grows up rapidly. Users require searching not only by specifying values of attribute-like data, such as file name, creation time and keyword, but also by the content of their data. For example, the user sees a cathedral while watching a movie but he or she is not sure whether he or she visited that place or not. Using this snapshot a private collection of holiday photos can be searched for images containing that cathedral. In theory, it is not sufficient to store data and search them by exact match but rather by means of similarity, i.e. retrieving data items similar to a query item. Similarity searching is especially requested in multimedia databases [1], digital rights management, copy detection, computer aided diagnosis [2], astronomy [3], biology [4,5], chemistry [6,7], and psychology (e.g., factor analysis, cluster analysis, multidimensional scaling, Shepard's generalization model, and various contrast models). In these fields, theoretical primal background for querying or questioning by similarity is defined.

In this paper, we focus on distance measures and similarity functions applied to audiovisual data and present a way of adopting the idea of property sharing – contrast models – for the approximate similarity searching. In practice, we propose a solution to retrieving database images that contain an object captured in a query image. In general, an audiovisual data item (object) is preprocessed and a high number of so-called features is extracted. Each feature describes a region of the object exactly or in a simplified way. We assume the following properties hold: (a) described regions can overlap one another, but they must not cover the whole object; (b) features are addressable within the object (in any suitable coordinate system). Such view is satisfied by SIFT [8] or SURF [9] local descriptors, for example.

There are two main contributions of this paper. Firstly, it presents the new similarity function called *proximity-based order-respecting intersection* and denoted as ε -intersection. For a query characterized by a set of features, this function finds the most similar objects that contain as many similar features as possible. This similarity function could be used directly as the function for evaluation of distance between two multimedia objects, or as a basic component for forming complex similarity functions, e.g., à la Jaccard index/coefficient [10,11] and Tversky’s feature contrast model [12]. Secondly, a new retrieval set reduction method based on ε -intersection is proposed. This method can reduce the amount of features processed and compared during querying, so processing time is decreased.

For simplicity of the formal definitions we assume that database images contain the query images only once (the query object does not repeat within an object). Our approach can be slightly modified to handle also this situation, but even the solution presented in this paper is capable of answering the question “Is this query image contained in an image in the database image?”.

2 Related Work

Searching for sub-images in an image database and copy-detection of images have been studied by various researches, mostly from the computer vision groups. They usually focus on the efficient image representation, indexing techniques, and ways of defining queries rather than on data organization. The most common approach discussed in literature proceeds from so-called information reduction techniques. This approach focuses on the compressing the space needed to encode one feature. This is done by a dimensionality reduction technique, e.g., locality sensitive hashing (LSH) and k-means clustering.

In [13], the authors present a solution to storage implementation of the LSH-coded descriptors that allows searching for sub-images in linear time. However, this implementation does not handle the spatial position of features. Another retrieved information reduction [14] uses the properties of PCA-SIFT descriptors [15] directly. In particular, their hierarchical ordering and bit representation of each feature are used. This leads to the most-significant bit index files, which are memory-oriented structures storing bit prefixes of PCA-SIFT descriptors.

Neither this technique indexes the spatial information. In [16], the geometric min-hash (GmH) algorithm is proposed. It is based on the original min-hash [17], but it incorporates the spatial context of features. From the searching point of view, it identifies regions in an image in which the identical or almost-identical groups of features with respect to the query image occur. It eliminates the features that are encoded to one visual word, which can cause problems when processing repeating patterns or when the queried sub-image occurs in the original image multiple times. By analogy to our approach, GmH optimizes the retrieval process by selecting some important features (anchors). However, only small spatial surrounding is considered instead of taking into account spatial order of features. Spatial order offers better rotation invariance. GmH is limited to small query images. At last, the method presented in [18] finds small logos in a natural image collection. SIFT features are reduced using the multi-probe locality sensitive hashing [19]. To determine the geometrically close features the RANSAC algorithm [20] is applied. The similarity score is eventually evaluated using an affine transformation model.

In contrary to the approaches mentioned above, the authors of [21] does not use local descriptors for the searching, but global features taken from MPEG-7 specification. Namely the edge-histogram and the color layout descriptors are used. Images in the database as well as the query image are segmented to chunks of pre-defined size and the mentioned features are extracted. Searching procedure then finds correspondence between the database images and the query image. For good retrieval results of sub-images segments of an image must overlap significantly, so this technique must be tuned properly. For this reason, we favour the local descriptors.

If a distance measure based on local descriptors is defined, it can also be used in different similarity searching models and can be applied in various applications, such as in architecture [22] or speech recognition [23].

3 Proximity-Based Order-Respecting Intersection

We represent real-world objects (texts, pictures, audio samples, etc.) as *entities* where each entity has a unique *identification* and is *characterized* by a set of *features*. We assume that the number of features can vary in entities, i.e. depending on the content of the entity, the number of extracted features may be different. Moreover, we distinguish two parts of each feature: the *locator* and the *descriptor*. The form and representation of locators and descriptors conform to a chosen system. Usually, the locator is expressed as a low-dimensional vector in some generally-used coordinate system. A descriptor can be a SIFT [8] or SURF [9] local descriptor, but we are not limited only to them. In general, most of the feature types employed in computer vision satisfy these requirements. In addition, our presented method handles also similarities between textual data, video sequences or 3-dimensional models without any modifications.

To compute similarity between two entities, we have to compare the corresponding sets of features. This is usually solved by evaluating intersection of

the sets. The traditional mathematical definition of intersection compares the elements on equality (i.e. the descriptors must be identical), which is not sufficient for our needs. Thus, we consider two elements as one if and only if their descriptors are “close”. Moreover, the positions of elements should be respected, i.e. the locators of near elements should be “close” too. Adopting the approach of multi-sets and computing intersection over them would be promising but the multi-set resulting from such intersection does not contain any information about the original elements. Moreover, the issue of combining close elements into one should be addressed. This may be solved by taking one of the close elements as the representative in the result or computing a new, possibly “middle”, element of them. Nonetheless, the middle element cannot be generally obtained correctly due to its complex internal structure, e.g., in metric spaces we are not able to “compute” a new object. Also taking only one representative might not be descriptive enough. For the same reason, the approach based on fuzzy sets [24] cannot be applied. As a result, a special solution that constructs a theoretical framework and allows an efficient implementation at the same time is needed.

Before presenting the proposed solution to compare entities, we introduce necessary formal definitions of the feature and entity.

Definition 1 (Feature). Feature f is an ordered k -tuple, $f = (p_1, p_2, \dots, p_k)$, $0 < k < \infty$ of attributes (components) so that $\forall i : p_i \in \langle 0, 1 \rangle \subset \mathbb{R}$. The first d attributes form the locator of the feature, i.e. the locator is a d -dimensional vector (p_1, p_2, \dots, p_d) . The remaining $k - d$ attributes form the descriptor, i.e. (p_{d+1}, \dots, p_k) . The set of features f having the same structure, i.e. the same size and meaning of locator and descriptor, is denoted as F .

For exposition purposes, we use $f | p_i$ to denote the attribute p_i of the feature f . While $f | \text{Loc}$ and $f | \text{Desc}$ denote the locator and descriptor of f , respectively.

Definition 2 (Entity). Let \mathbb{E} be a finite set of vectors (ordered tuples) of F of the length one or two, such that $\mathbb{E} = \{F_i | F_i \in \mathcal{P}(F)^k, k = 1 \vee k = 2\}$, where $\mathcal{P}(F)$ is the power set of F . We denote the length of the vector F_i as $|F_i|$. We denote the j 's constituent (vector element) of F_i as F_i^j . The value of F_i^j could be “undefined” if $j > |F_i|$. We also define that $f_i \in F_k$ for any $F_k \in \mathbb{E} \Leftrightarrow f_i = F_k^j$ for any $1 \leq j \leq |F_k|$. If $\forall F_k \in \mathbb{E}$ and $\forall f_i \in F_k$ there does not exist any $F_m \in \mathbb{E}$, $k \neq m : f_i \in F_m$, then \mathbb{E} is called entity.

In particular, this definition allows an entity to consist of one- or two-element vectors. The two-element vector represents a pair of close features, i.e. the features considered as one by an intersection function (see the next section for details). An entity representing an audio-visual object contains only one-element vectors, e.g., SIFT local descriptors. For simplicity, $\mathcal{P}(F)$ will denote the system of entities from now and on.

Definition 3. Let \mathbb{E} be an entity such that $\mathbb{E} = \{F_1, F_2, \dots, F_n\}$, $n = |\mathbb{E}|$ and $\forall F_i : |F_i| = 2$. Then we define the function $\sigma_\alpha^i : \mathcal{P}(F) \mapsto \mathbb{R}^n$, $\sigma_\alpha^i(\{F_1, F_2, \dots, F_n\}) = (F_1^i | p_\alpha, F_2^i | p_\alpha, \dots, F_n^i | p_\alpha)$, $i \in \{1, 2\}$; p_α is the α^{th} component of the feature F^i .

Definition 4. We define π as the permutation of \mathbb{R}^n as follows: $\pi = (r_{i_1}, r_{i_2}, \dots, r_{i_n})$ ($r_{i_j} < r_{i_{j+1}}$) \vee ($r_{i_j} = r_{i_{j+1}} \wedge i_j < i_{j+1}$). We also define $\pi^{-1}(r_i)$ as the identification of the position of element r_i in the permutation π .

Such permutation can be used to construct a function which measures the distance between two vectors of natural numbers. This function, called *ord*, takes two permutations $\pi(R)$, $R = (r_1, \dots, r_n)$, and $\pi(S)$, $S = (s_1, \dots, s_n)$, and evaluates the difference between them, i.e. $ord : \mathbb{N}^n \times \mathbb{N}^n \mapsto \mathbb{R}$. For example, the *ord* function can be defined as follows:

- The number of permuting positions:

$$ord(\pi(R), \pi(S)) = \sum_{i=1}^n \begin{cases} 0 & \text{if } \pi^{-1}(r_i) = \pi^{-1}(s_i) \\ 1 & \text{otherwise} \end{cases}$$

- Spearman's Footrule, the absolute value of differences between positions:

$$ord(\pi(R), \pi(S)) = \sum_{i=1}^n |\pi^{-1}(r_i) - \pi^{-1}(s_i)|$$

- Spearman's Rho;
- Kendall's Tau;
- and others.

Having the definition of the *ord* function, we can define the *ORD* function evaluating the order error in all dimensions of entities' locators.

Definition 5 (ORD function). Let \mathbb{E} be an entity and *ord* be a function $\mathbb{N}^{|\mathbb{E}|} \times \mathbb{N}^{|\mathbb{E}|} \mapsto \mathbb{R}$. We define the function $ORD_\alpha : \mathcal{P}(F) \mapsto \mathbb{R}$ such that

$$ORD_\alpha(\mathbb{E}) = ord(\pi(\sigma_\alpha^1(\mathbb{E})), \pi(\sigma_\alpha^2(\mathbb{E}))),$$

Let the dimensionality of locator be d . Then the *ORD* function is defined as the weighted sum

$$ORD(\mathbb{E}) = \sum_{\alpha=1}^d w_\alpha ORD_\alpha(\mathbb{E}),$$

$w = (w_1, \dots, w_d) \in \mathbb{R}^d$ is a user-defined parameter.

In particular, the *ORD* function evaluates the number of well-ordered locators gradually for individual locator elements (dimensions). In other words, this function is used to filter out matching pairs of descriptors but which are not in the same order in the query and in a database object. Please remark that the spatial distribution of features is crucial to identify a correct match. For example, searching for sub-images based on comparing descriptors only leads to false positives in the query response, i.e. images containing features similar to one in the query image but differently positioned, so representing a completely different object than the one captured in the query.

Finally, we define the ε -intersection that identifies pairs of features between two images that are similar (in their descriptors) and well-ordered (in their locators).

Definition 6 (ε -intersection). Let the parameter ε and the functions δ and ORD are defined. For any two entities $\mathbb{A}, \mathbb{B} \in \mathcal{P}(F)$ the ε -intersection function \cap_ε is defined as $\mathbb{A} \cap_\varepsilon \mathbb{B} = \{(a_i, b_j) \mid a_i \in \mathbb{A}, b_j \in \mathbb{B}\}$ if $\forall a_i \in \mathbb{A}, b_j \in \mathbb{B}$ the following holds:

$$\begin{aligned}
 (a_i, b_j) \in \mathbb{A} \cap_\varepsilon \mathbb{B} &\Leftrightarrow \\
 \delta(a_i \mid \text{Desc}, b_j \mid \text{Desc}) &\leq \varepsilon && \wedge \\
 \neg \left(\exists a_k \in \mathbb{A} : \delta(a_i \mid \text{Desc}, b_j \mid \text{Desc}) > \delta(a_k \mid \text{Desc}, b_j \mid \text{Desc}) \wedge \right. \\
 \left. \text{ORD}(\{\dots, (a_i, b_j), \dots\}) > \text{ORD}(\{\dots, (a_k, b_j), \dots\}) \right) &&& \wedge \\
 \neg \left(\exists b_m \in \mathbb{B} : \delta(a_i \mid \text{Desc}, b_j \mid \text{Desc}) > \delta(a_i \mid \text{Desc}, b_m \mid \text{Desc}) \wedge \right. \\
 \left. \text{ORD}(\{\dots, (a_i, b_j), \dots\}) > \text{ORD}(\{\dots, (a_i, b_m), \dots\}) \right) &&&
 \end{aligned}$$

In details, the parameter ε controls the degree of similarity of features' descriptor, i.e. the threshold for taking two descriptors as an “identical” one. If there are two such close descriptors, their corresponding locators are then checked to be in the same order (call to function ORD). This check can be easily enriched with a threshold too. At last, notice that the pairs identified by the intersection are returned as a new entity.

The ε -intersection is not deterministic. We can get more results for the same pair of entities. Assume that for one particular feature $a_i \in \mathbb{A}$ there are two other features $b_{j_1}, b_{j_2} \in \mathbb{B}$, $b_{j_1} \neq b_{j_2}$ such that $\delta(a_i \mid \text{Desc}, b_{j_1} \mid \text{Desc}) = \delta(a_i \mid \text{Desc}, b_{j_2} \mid \text{Desc}) \leq \varepsilon$ and at the same time $\text{ORD}(\{\dots, (a_i, b_{j_1}), \dots\}) = \text{ORD}(\{\dots, (a_i, b_{j_2}), \dots\})$. The definition is fulfilled for either b_{j_1} or b_{j_2} , but there is no special instruction which feature to prefer. However, the our implementation is deterministic since features of each entity are stored in a list, so the order in which features are checked is always the same.

From the efficiency point of view, this definition of ε -intersection is very demanding in terms of computational resources. Average computation complexity of an algorithm is quadratic, but in the worst case it can be exponential. Fortunately, we usually do not need to evaluate the intersection of the whole large sets but good results are obtained by taking a small subset of features (or a small image). In the following section, we present a possible algorithm optimized by using selected features, called anchors, only. This trick is inspired by optimizations done in [16].

4 Anchors

Local descriptors extracted from the multimedia, textual or another scientific data usually contain the “scale” attribute. This attribute represents the importance of the descriptor (e.g., the spatial size of the feature surroundings). This leads to the idea that we do not need to compute the ε -intersection for the whole set of features, but for many applications it is sufficient to evaluate the



Fig. 1. Example of bounding rectangles: For the query (left) a database image (right) has been retrieved. The black rectangle covers the anchors identified. The red rectangle is the expansion of the black one using spatial information from the query image.

ε -intersection on few most-important features, which we call *anchors*. The procedure cannot filter out any matching objects, thus it can introduce some false positives. They can be filtered out using the complete sets of features later if required.

In the following, we propose an algorithm for identifying anchors during the evaluation of ε -intersection. The input of this algorithm contains a query object (image) Q , the number *nof* of important features, the threshold ε on similarity of feature descriptors, and two limits on the number of features (*limit1* and *limit2*). The algorithm proceeds in the following steps:

1. Let $Result = \emptyset$ and $Temp = \emptyset$;
2. Extract local features from query entity Q and take *nof* most significant ones with respect to the scale attribute. Take *nof* features randomly if the scale attribute is missing. These features form a set $F_Q = \{q_1, \dots, q_{nof}\}$.
3. For each local feature $q_i \in F_Q$, find all local features f_k from the database such that $\delta(q_i | Desc, f_k | Desc) \leq \varepsilon$. The pairs (q_i, f_k) are added to the set $Temp$;
4. Group the local features in $Temp$ by their database entity identification. Thus a set of groups of features corresponding to the same entity is obtained. This set forms the preliminary result – $Result = \{G_{\mathbb{E}_1}, \dots, G_{\mathbb{E}_n}\}$, where $G_{\mathbb{E}_i} = \{(q_v, f_w), \dots, (q_x, f_y)\}$;
5. For each feature $q_i \in F_Q$, in each $G_{\mathbb{E}_j}$ keep only the pair (q_i, f_k) of features which are the most similar, i.e. $f_k = \operatorname{argmin}_{(q_i, b_l) \in G_{\mathbb{E}_j}} (\delta(q_i | Desc, b_l | Desc))$;
6. From $Result$ remove all groups containing less than *limit1* pairs;
7. For each dimension α of the feature locator and for each $G_{\mathbb{E}_i} = \{(q_v, f_w), \dots, (q_x, f_y)\}$, do:
 - (a) Sort the pairs by the value of $f_i | \alpha$;
 - (b) Remove from $G_{\mathbb{E}_i}$ all pairs with unsuitable order in $q_j | \alpha$ to maximize the longest possible sequence of candidates.
8. From $Result$ remove all groups containing less than *limit2* pairs again;
9. The remaining features in each $G_{\mathbb{E}_i}$ in $Result$ form the anchors for the database entity \mathbb{E}_i .

To localize the query in the entities in $Result$, we take the locator of each anchor and form the minimum bounding rectangle. This rectangle encloses the

match. Due to a different number of features taken from Q that have been identified in each resulting entity, the match can be smaller than the original query image. In Fig. 11, there is an example of a query object and a database image in which the query has been identified. The minimum bounding rectangle is emphasized with the black color. We can see that the sub-image identified does not correspond to the query image in its size, so we use spatial information to extend the match and provide the user with a better answer. This procedure is based on positions of features identified in both the database object and the query object. In particular, we take the features from the query that were used to find the match and the locators of these features form a region in the query object. We do the same for the database object. By comparing the extents of these regions, we can enlarge the region in the database object to have the same aspect ratio. The result of this procedure is depicted as the red rectangle in the figure. Moreover such extension gives to us a chance to evaluate the ε -intersection more precisely and also efficiently in order to prune more objects not containing the query.

5 Experimental Results

In this section, we present experimental results obtained by evaluating the ε -intersection on two real-life datasets. The first one is a small collection of private photos taken during sightseeing all around Europe. The second one contains thousands of logos of companies. From these datasets, we extracted SIFT [8] and SURF [9] features and applied the proposed sub-image search algorithm. In experiments, we focus on three phenomena. Firstly, the effectiveness of the proposed algorithm is studied on the first data set. Secondly, we tackle the issue of ranking the objects in result by the *ORD* function. Finally, we study the algorithm's parameters, namely ε , *nof* and *limit1* and their influence to performance indicators. These two last experiments are undertaken using the logo data set.

5.1 Architectural Monuments

In this group of experiments, we measure recall and precision if the number of anchors identified changes. We used the first dataset which consists of 2,000 images (1,152 by 1,204 pixels each) of architectural monuments. From this collection, we picked four sub-images and manually localized the correct result (ground truth), which always contained eight images (i.e. just these images contained the query as their sub-image). The parameters of the proposed intersection algorithm were set as follows. The number of the most significant features taken from the query (*nof*) was not fixed but all features having the scale property greater than four pixels were taken instead. It resulted in using 100 features on average. In this collection, the number of local features extracted was quiet high, so we require to filter out entities having just very little similarity. Thus *limit1* was set to eight. As the distance function δ for comparing SIFT and SURF descriptors, we use the Euclidean distance. So the value of ε was fixed to 240 and 0.06, respectively.

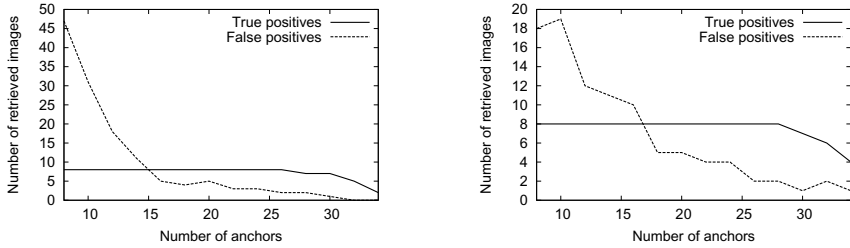


Fig. 2. The number of matching images (true positives) and the number of non-matching images (false positives) in the query result while at least the specific number of anchors is found

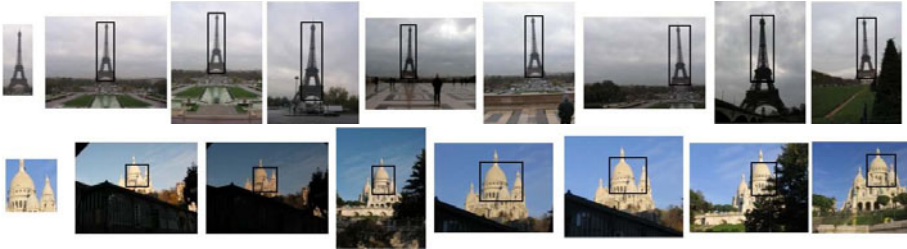


Fig. 3. Results of two queries exposing true positives only. The minimum bounding rectangles of anchors identified in each image are emphasized. The query image is the left-most image in the row.

They represent empirically verified values. For the last parameter, the *ORD* function, we used the Spearman’s footrule of permutations with all the weights w_i equal to one.

Figure 2 depicts the number of *false positives* (FP) and the number of *true positives* (TP) retrieved from the dataset while the minimum number of anchors (*limit2*) was changing. Thus, the recall is $TP/8$ and the precision is $TP/(TP + FP)$, where eight is the ground-truth (explained above).

From the results on the left (SIFT features), we can read that the number of false positives (images not containing the query but retrieved) is decreasing rapidly as the *limit2* increases. For $limit2 > 31$, there are no non-matching images in the result set. Nonetheless, for these values, some matching images were filtered out, i.e. the number of true positives is decreasing. Almost identical behavior can be observed for SURF features (the right-hand figure). The good value of *limit2* is between 20 and 30 for which the precision is above 50% and recall is 100%. Notice that testing values of *limit2* less than eight makes no sense since *limit2* must be greater than or equal to *limit1* (please refer to the algorithm in Section 4 for explanation). An example of query results is given in Fig. 3 where the left-most image is the query and $limit2 = 15$.

5.2 Company Logos

The major disadvantage of the ε -intersection is that the result is not ranked in any way. It is an unordered set if we strictly follow the mathematical definition of a set. In these trials, we study the influence of the *ORD* function on ranking the result set. In the evaluation of ε -intersection, the following setting was used: the *ORD* function is the Spearman's footrule with weights set to one, $nof = 16$, $\varepsilon = 240$, *limit1* and *limit2* were both set to 6. We selected such low values by purpose because we pursued some false positives in results to show the effect of ranking. The *ORD* function was then used to rank the objects in the response.

In this group of experiments, we used a collection of 15,535 logos of companies, services and products. This dataset has been selected for more reasons. Firstly, the logo images form a compact data set with respect to image size and color. Secondly, the ground-truth of the data set was available. Lastly, similarity search algorithms based on global image features and previously developed within the MUFIN (Multi-feature indexing network) project¹ were available for a direct comparison with local-feature-based approaches. From each image in the collection, SIFT local features were extracted – 152 features were obtained per image on average. Altogether we had to organize 2,359,839 features.

The proposed algorithm for finding sub-images was implemented within the MUFIN project using the MESSIF framework [25], so we had an advantage of using index structures for similarity search in feature descriptors. In particular, we built the M-tree [26], but any other metric index structure, even distributed one, could have been applied.

For the demonstration purposes, we used two queries only which got selected in the following way. We picked 37 images from the collection at random and cut out typical parts of logos in these images. Then, the ε -intersection algorithm was evaluated, and two queries returning the most objects were selected. Namely, they are the “ČKD” and “Feron” companies. Before returning an answer to the user, ranking on the result was done by the following distance function:

$$d(Q, O) = \frac{ORD(Q \cap_{\varepsilon} O)}{|II|},$$

where $|II|$ is the length of permutation used in *ORD* function. This ranking procedure measures the number of disordered features as compared to the their order in the query object. The division factor was used to make these numbers comparable because images in the response set can have a different number of features matching the query.

Figures 4 and 5 depict the ranked results for the selected queries. The query image is always the top-left image and the rectangles show the bounding rectangles of locators of matching features. In case of queries, the rectangle bounds the sixteen most important features used to compute the ε -intersection. As for the ČKD logo, the response set contains all the matches and no false positives are present. This was verified by checking the data collection manually. Thus, the

¹ <http://mufin.fi.muni.cz/>



Fig. 4. Example of searching for a sub-image: the ČKD logo



Fig. 5. Example of searching for a sub-image: the Feron logo

rank here is not very important. The situation is much more interesting in case of the Feron logo, where some false positives are present. The ranking function assigned these non-matching objects low ranks, so most of the relevant images got the highest rank. But the three logos that contain the query as a sub-image were not ranked properly because the ϵ -intersection identified a few features in the image that were not localized within the sub-image containing the logo. This is caused by the fact that the definition of ϵ -intersection primarily focuses on similarity between feature descriptors and secondarily on the corresponding locators. Thus, in these images, there were some out-of-the-sub-image features more similar than the ones within the sub-image.

5.3 Influence of Algorithm Parameters

Finally, we studied the influence of algorithm’s parameters to the response from both the effectiveness (recall and precision) and efficiency points of view. The efficiency was measured as the time to complete query evaluation and as the

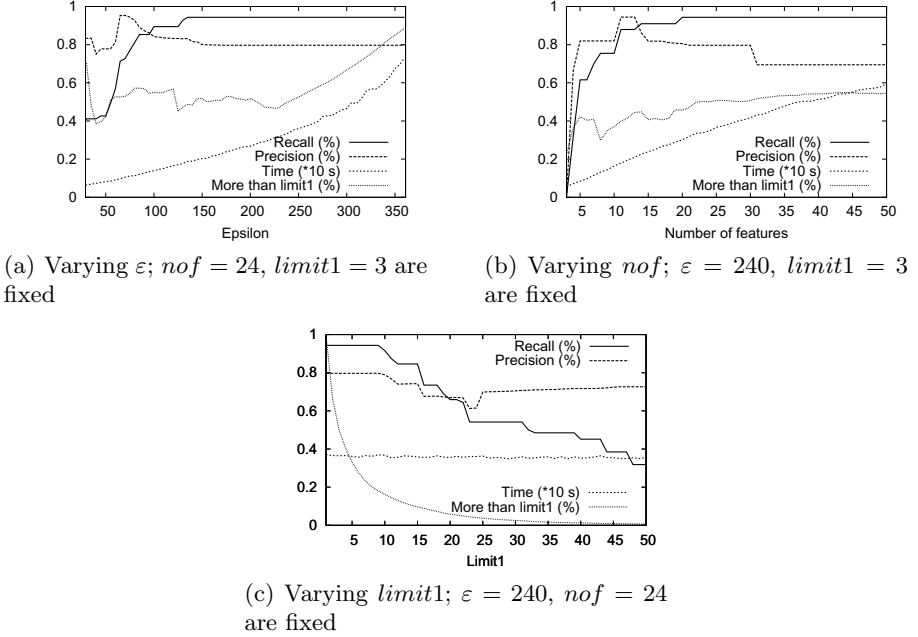


Fig. 6. Influence of parameters to the algorithms’s results (recall, precision, computation time and ratio of objects removed from the *Result* set). Two of three parameters (ϵ , nof and $limit$) are fixed at each graph, one of them varies.

percentage of objects having more than $limit1$ matching features (Step 6 of the algorithm in Section 4). We focused on the parameters ϵ , nof and $limit1$ and conducted three experiments. In each of them, we varied one of these parameters while retaining the others fixed.

In Fig. 6, the average values of the measured indicators obtained for queries having at least six relevant images (i.e. the ground truth ≥ 6) are depicted. For exposition purposes, we normalized the run time to the interval $[0, 1]$, where one represents 10 seconds. The line “More than $limit1$ ” denotes the ratio of the number of objects having more than $limit1$ features to the number of all objects entering Step 6 of the algorithm. To ensure objectivity of the presented run time, we executed the range queries in Step 3 of the algorithm sequentially.

Firstly, we varied ϵ and fixed nof and $limit1$ to 24 and 3, respectively. As Fig. 6(a) depicts, ϵ should be at least 150 for the SIFT features and this data set to obtain 100% recall. This also confirms the general scientific experience that the maximum distance of relevant SIFT features is around 240. Higher values of ϵ may only increase the number of false positives. The results are much more influenced by the value of nof , which is the core of the experiment in Fig. 6(b). We fixed ϵ to 240 and $limit1$ to 3. The recall is increasing as the number of features taken from a query increases. However, taking all the features extracted from the query may lead to worse results in precision since we would also take

features that are not important. Consequently, they can introduce many false positives in the response. From the efficiency point of view, the value of *nof* induces the number of range queries executed, so higher values lead to high CPU costs. A good value for *nof* is about 24. In the last experiment, we fixed ε and *nof* to the values verified so far, i.e. to 240 and 24, respectively. Figure 6(c) shows results for varying value of *limit1*. The run time depicted is constant, which confirms that the major computational demands are in evaluating range queries in Step 3 of the algorithm (please recall that *nof* and ε are fixed). On the other hand, the expected behavior is that recall and precision decreases if *limit1* increases. This is caused by filtering out many objects in Step 6 of the algorithm simply because that not all features out of *nof* are found in all retrieved images. Moreover, setting *limit1* to values greater than *nof* usually leads to very low recall otherwise each of *nof* features should be paired with more than one feature in each image, which is not very likely for a query feature having a high value of scale property. This phenomenon can also be observed in Fig. 2. To sum up, a good values of *limit1* are relatively small (< 10), because they can filter out just “random” matches, e.g., images that contain one or two matching features.

6 Conclusions and Future Work

In this paper, we have proposed a proximity-based order-respecting intersection and applied it to image databases. This intersection is capable of sub-image search, i.e., images containing an object captured in the query image are retrieved from the database. An advantage of the proposed intersection is that to some extent it is invariant to the scale. In particular, the query image is found also in database images that contain the query but scaled. The definition of ε -intersection is very general and it is applicable not only to visual data but also to other data domains, such as audio data and textual data, where searching for “sub-patterns” makes sense. In addition to that, we have shown that our formal and general definition provides a large degree of variability and possibilities of customization. For example, several options to specify the user-defined *ord* function were given. From the data point of view, our algorithm is not fixed to the high-dimensional representation of descriptors only, but various reduction techniques applied to the features extracted (such as visual words and locality sensitive hashing) can be utilized to increase the efficiency of the proposed solution. In case of visual words, the database would consist of IDs of visual words representing a group of original features, so the range queries needed to retrieve a candidate set of images would reduce to a primary-key search. The other parts of the algorithm would be left unmodified.

Apart from the formal definition, we have presented an idea of implementation of our algorithm in Section 4. We conducted three groups of experiments on two real-life data sets. In these trials, we mainly studied the issue of effectiveness, i.e., whether the proposed algorithm finds images containing the query or not. However, we have also presented the algorithm’s costs expressed in running time, to give a view how demanding the proposed solution is. From this performance study, we can conclude that the ε -intersection is a promising concept.

The major problem of using local descriptors such as SIFT or SURF is that the extractors produce tens or even hundreds of features, which leads to low performance because the database of extracted features grows by two or three orders of magnitude. This consequently slows down the search for features similar to one in the query. Thus, in the future, we plan to analyze and optimize the efficiency of this algorithm in a way that it can be applied to large databases containing millions of data items or even more. Also the influence of different *ORD* functions should be tackled. Next, we would like to extend the definition of ε -intersection to allow defining either Jaccard coefficient or Tversky's feature contrast model. At last, the issue of identifying all occurrences, not just one, of the query in a matching database item.

Acknowledgments. This research was supported by national projects GACR P103/10/0886, GACR GA201/09/0683 and MUNI/A/0922/2009. The access to the METACentrum (super)computing facilities provided under the research intent MSM6383917201 is (highly) appreciated/acknowledged. We used this infrastructure for feature extraction. We also thank the LogoOpen company for providing the logo data set.

References

1. Batko, M., Dohnal, V., Novak, D., Sedmidubsky, J.: MUFIN: A Multi-Feature Indexing Network. In: Proceedings of the 2nd International Conference on Similarity Search and Applications (SISAP 2009), pp. 158–159. IEEE Computer Society, Los Alamitos (2009)
2. Petrakis, E.G.M., Faloutsos, C.: Similarity searching in medical image databases. *IEEE Trans. on Knowl. and Data Eng.* 9, 435–447 (1997)
3. Brunner, R.J., Djorgovski, S.G., Prince, T.A., Szalay, A.S.: Massive datasets in astronomy, pp. 931–979. Kluwer Academic Publishers, Norwell (2002)
4. Hubálek, Z.: Coefficients of association and similarity, based on binary (presence-absence) data: An evaluation. *Biological Reviews* 57, 669–689 (1982)
5. Sneath, P.H.A., Sokal, R.R.: *Numerical Taxonomy: The Principles and Practice of numeric Classification*. W. H. Freeman and Company, San Francisco (1976)
6. Monev, V.: Introduction to similarity searching in chemistry. In: *Match-Communications in Mathematical and in Computer Chemistry*, vol. 51, pp. 7–38. Bulgarian Academy of Sciences (2004)
7. Flower, D.R.: On the properties of bit string-based measures of chemical similarity. *J. Chem. Inf. Comput. Sci.* 38, 379–386 (1998)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60, 91–110 (2004)
9. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Comput. Vis. Image Underst.* 110, 346–359 (2008)
10. Jaccard, P.: Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 241–272 (1901)
11. Zezula, P., Amato, G., Dohnal, V., Batko, M.: *Similarity Search - The Metric Space Approach*, vol. 32. Springer, Heidelberg (2006)
12. Tversky, A.: Features of similarity. *Psychological Review* 84, 327–352 (1977)

13. Ke, Y., Sukthankar, R., Huston, L., Ke, Y., Sukthankar, R.: Efficient near-duplicate detection and sub-image retrieval. In: *ACM Multimedia*, pp. 869–876 (2004)
14. Roth, G., Scott, W.: Efficient indexing for strongly similar subimage retrieval. In: *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision (CRV 2007)*, pp. 440–447. IEEE Computer Society, Washington, DC, USA (2007)
15. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR 2004)*, pp. 506–513. IEEE Computer Society, Los Alamitos (2004)
16. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 17–24. IEEE Computer Society, Los Alamitos (2009)
17. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007)*, pp. 549–556. ACM, New York (2007)
18. Joly, A., Buisson, O.: Logo retrieval with a contrario visual query expansion. In: *Proceedings of the seventeen ACM International Conference on Multimedia (MM 2009)*, pp. 581–584. ACM, New York (2009)
19. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe lsh: efficient indexing for high-dimensional similarity search. In: *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007)*, VLDB Endowment, pp. 950–961 (2007)
20. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
21. Ryu, M.S., Park, S.J., Won, C.S.: Image retrieval using sub-image matching in photos using MPEG-7 descriptors. In: Lee, G.G., Yamada, A., Meng, H., Myaeng, S.-H. (eds.) *AIRS 2005*. LNCS, vol. 3689, pp. 366–373. Springer, Heidelberg (2005)
22. Zhang, W., Košecká, J.: Hierarchical building recognition. *Image Vision Comput.* 25, 704–716 (2007)
23. Hazen, T.J., Saenko, K., La, C.H., Glass, J.R.: A segment-based audio-visual speech recognizer: data collection, development, and initial experiments. In: *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI 2004)*, pp. 235–242. ACM, New York (2004)
24. Santini, S., Jain, R.: Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 871–883 (1999)
25. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: Thanos, C., Borri, F., Candela, L. (eds.) *Digital Libraries: Research and Development*. LNCS, vol. 4877, pp. 1–10. Springer, Heidelberg (2007)
26. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB 1997)*, pp. 426–435. Morgan Kaufmann, San Francisco (1997)

Experiences with Shape Classification through Fuzzy c -Means Using Geometrical and Moments Descriptors

Ugo Erra¹ and Sabrina Senatore²

¹ Università della Basilicata, Dipartimento di Matematica e Informatica,
Viale Dell'Ateneo 10, Macchia Romana - 85100, Potenza, Italy
ugo.erra@unibas.it

² Università di Salerno, Dipartimento di Matematica e Informatica
Via Ponte Don Melillo - 84084 Fisciano(SA), Italy
ssenatore@unisa.it

Abstract. Due to the growing diffusion of digital media, most of real world applications have data with multiple modalities, from multiple sources and in multiple formats. The modelling of information coming from multimedia sources represents an important issue for applications which achieve multimedia mining activities. In particular, the last decades have witnessed great interest in image processing by “mining” visual information for objects recognition and retrieval. Some studies have revealed the image disambiguation based on the shape produces better results than features such as color or texture; moreover, the classification of objects extracted from an image database appears more intuitively formulated as a shape classification task.

This paper presents an approach for 2D shapes classification. The approach is based on the combined use of geometrical and moments features extracted by a given collection of images and achieves shape-based classification exploiting fuzzy clustering techniques.

1 Introduction

In the age of digital information, the growing amount of large-scale image repositories in many application domains emphasize the need for effective means for mining and classifying digital image collections.

In general, two different approaches have been applied to allow image retrieval: one based on textual information whereas the other based on image content information. The first retrieval approach consists of attaching textual metadata to each image and then submits a keyword-based query to the database in order to retrieve them [23]. This approach requires an initial annotation activity which often results laborious and time-consuming; moreover, it is a human driven process thus, similar images characteristics can be expressed by different users with different terminologies, affecting the performance of the keyword-based image search.

For these shortcomings, (semi-)automatic approaches have been achieved to process the image in order to get more “objective” *content-based* image properties such as color, texture, and shape. Content-Based Image Retrieval (CBIR) systems involve characterizing an image using a set of features; retrieval or classification is then performed by measuring similarity to a required query image [34] contrasting to the effort needed to annotate images.

Images can be particularly complex to manage; thus, CBIR techniques require the translation of high-level user perceptions into low-level image features. To cope with the so called “semantic gap” problem, these features should be consistent and invariant to remain representative for the images collection in a database. Image indexing is not an issue of string processing (as in the case of standard textual databases), but an n-dimensional vector describes the characteristic of the image [14]. Then, the image retrieval process consists of discovering all the images whose features are similar to the query example image. A direct drawback is that these low-level image features are often too restricted to describe images on a conceptual or semantic level, impacting on the performance of image retrieval approaches.

On the other hand, the CBIR technology tries to address two intrinsic problems: (a) how to mathematically describe an image, and (b) how to assess the similarity between a pair of images based on their abstracted descriptions. Recent methodology development employs statistical and machine learning techniques in various aspects of the CBIR technology. In image classification methods, the approaches are based on learning-based classification and non-parametric classifiers. As been pointed out in [6], despite the large performance gap between non-parametric classifiers and state-of-the-art learning-based, the non-parametric image classification have been considerably under-valued and offer several advantages: (i) can naturally handle a huge number of classes; (ii) avoid overfitting of parameters, which is a central issue in learning based approaches; (iii) require no learning/training phase. As explained later in this paper, our approach could be considered parameter-free, when the number of cluster is known a priori. The focus of this work is to define an approach for image classification and retrieval based on 2D shape features, exploiting fuzzy clustering techniques. The paper is organized as follows. Section 2 gives a sketched overview of the related works in this area, then as a background, Section 3 focuses on the image processing for the image analysis and features extraction whereas Section 4 introduces the fuzzy clustering algorithm exploited in this approach. Finally, Section 5 describes the experiments and provides the results. Conclusions and future works close the paper.

2 Related Works

Improvements in data storage and image acquisition technologies require new computer-assisted image understanding tools which support the large-scale image and media content datasets and provide assistance in image processing, query and retrieval. CBIR systems address these important issues in computer

vision and multimedia computing, supporting effective searching and browsing of large image digital libraries based on automatically derived image features [14]. Some examples of popular CBIR systems are QBIC, Virage, RetrievalWare, Photobook, Chabot, VisualSeek, WebSeek, MARS system, SurfImage, Netra, and CANDID (for additional details about them refer to [28]). Furthermore, a complete and exhaustive survey on CBIR developments and advances is provided in [9]. Almost all of these approaches are based on indexing imagery in a feature space. A feature represents a certain visual property of an image, either globally for the entire image or locally for a small group of pixels. The feature extraction is often considered as a preprocessing step, which represents the inputs to subsequent image analysis tasks. Typical features are color, texture, shape and region.

Also the increasing diffusion of images compression requires challenging techniques to extract visual features [22]: sophisticated global features such as the wavelets [29] and large collection of local image descriptors as SIFT [24].

Some other techniques improve the effectiveness of image retrieval through multi-features combination [15], [33] and then, by measuring similarity to a required query image [34]. Combination of words and features characterize annotated training sets of images [27], which will be used for classification or retrieval. In [13] a hierarchical feature subset selection algorithm for semantic image classification is defined, where the feature subset selection procedure is seamlessly integrated with the underlying classifier training procedure.

The image description and the user's perception of these features evidence the imprecise nature of the retrieval which can benefit by fuzzy techniques. Fuzzy logic is suitable for expressing queries which involve concepts and linguistic expression by means of fuzzy values rather than crisp features values [20].

Applying fuzzy processing techniques to CBIR approaches has been extensively investigated in literature. In general, fuzzy retrieval models offer more flexibility in the representation of the terms' index, preferences among terms in a query and ranked results. In particular CBIR models take advantage by using technique based on fuzzy theory for knowledge representation, for uncertainty management, against traditional information retrieval models based on boolean, vector-based or probabilistic representation. An example is given in [2] where a fuzzy information retrieval model for textual data has been extended to implement a model in image context. In [21], fuzzy logic has been employed to interpret the overall color information of images: according to the human perception, nine classes of colors are defined as features.

A fuzzy color histogram approach in [30] allows the evaluation of similarity through fuzzy logic-based operations. In [8], instead the similarity of two images has been defined by considering the overall similarity between families of fuzzy features. More specifically, each image has been associated to a family of fuzzy features (fuzzy set) representing color, texture, and shape properties. This approach reduces the influence of inaccurate segmentation, compared with other similarity measures based on regions and with crisp-valued feature representations.

Many CBIR approaches exploit clustering for preprocessing activities [12], specifically, fuzzy techniques are widely employed in image classification methods. In [25], a method to calculate image similarity measure using fuzzy partition of the HSI color space has been presented. In particular, the fuzzy c-means (FCM) clustering algorithm [5] has been shown to provide effective partitions for image segmentation on medical images [16], satellite images [18] [32], etc. Some extension and modification of FCM are applied to image segmentation in infrared images domain [19]. In [26] a modified version of FCM has been proposed, to solve the problem of large-scale image retrieval and classification, even though the clustering step is performed in lower-dimension space, and image retrieval is only performed in clustered prototypes. Yet, in the most of approaches, the execution time of the clustering algorithm is a critical point, which finds a solution in [19].

3 Design of the Feature Space

The first step toward the shape analysis of a given image involves separating the object (or region) of interest from other non-important image structures by using an image segmentation approach. There are several approaches for the extraction of the shape from a given image based on clustering methods, histogram methods, edge detection, level set methods, graph partitioning methods and so on. In general there is no a general solution and there is always an image where an approach does not yield good result, i.e., if the foreground and background share many similar colors, an approach could give a result with parts of background labelled as foreground object. This is challenging in shapes classification because any approach must take into account this drawback. In our implementation, we adopt the k-means clustering algorithm for image segmentation which is suitable when the foreground and background colors contrast sufficiently with each other.

A shape descriptor is a set of numbers that are extracted from the region of interest in order to describe a given shape feature. Efficient shape features must present some essential properties such as identifiability, invariance, noise resistance, statistically independence and so on.

In this work, we adopt three types of such shape descriptions: geometric description, invariant moments and affine moments. The geometric features discriminate shapes with large difference. They are useful to eliminate false hits and usually are not suitable as single description, in fact they are combined with other shape descriptors to better discriminate shapes. The moment instead, represents a mathematical concept coming from the concept of moment in physics. It is used in computer vision for both contour and region of a shape. In particular, the invariant moments [17] are one of the most popular and widely used contour-based shape descriptors. Affine moments invariants are instead features computed from moments that do not change their value in affine transformation.

In the case of geometric features, let P and A denote the shape perimeter and area, respectively. Note that perimeter and area are invariants respect to

translation and rotation but when combined, they are not invariant with respect to scale. The features we adopt are:

- Eccentricity E is the measure of aspect ratio. It is defined as the ratio $E = W_{bb}/H_{bb}$ where W_{bb} and H_{bb} are, respectively, the width and height of minimal bounding rectangle of the shape.
- Rectangularity R represents how rectangular a shape is, i.e. how much it fills its minimum bounding box. It is defines as $R = A/A_{bb}$ where A_{bb} is the area of the minimum bounding rectangle.
- Compactness C is a measure that combines area with perimeter. It is defined as $C = L^2/4\pi A$.
- The value π_{gen} is a measure of the compactness of a shape respect to a circle. It is defined as $\pi_{gen} = P/W_{bb}$.

Among the region-based descriptors, invariant moments m_{pq} are the simplest and is given as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad p, q = 0, 1, 2, \dots$$

where $f(x, y)$ is the intensity function at position (x, y) in a 2D gray level image. In order to obtain translation invariance, the central moments μ_{pq} should be applied:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad p, q = 0, 1, 2, \dots$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. Given central moments we are able to compute a set of 7 invariant moments [17], given by:

$$\begin{aligned} I_1 &= \eta_{20} + \eta_{02} \\ I_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ I_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} - \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2] + \\ &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[(3\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] \\ I_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] + \\ &\quad 4\eta_{11}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} - \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2] + \\ &\quad (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})[(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned}$$

where $\eta_{pq} = \mu_{pq}^\gamma$ and $\gamma = 1 + (p + q)/2$ for $p + q = 2, 3, \dots$. These moments are simple to calculate and they are invariant to translation, rotation and scaling but have an information redundancy drawback since the basis is not orthogonal [7]. From central moments with a little computational effort we are able to obtain also an affine transform invariance which includes the similarity transform and in addition to that stretching and second rotation. We adopt affine moments as defined in [10] and given as:

$$AMI_1 = (\mu_{20}\mu_{02} - \mu_{11}^2)/\mu_{00}^4$$

$$\begin{aligned}
AMI_2 &= (\mu_{30}^2 \mu_{03}^2 - 6\mu_{30} \mu_{21} \mu_{12} \mu_{03} + 4\mu_{30} + \mu_{12}^3 + \\
&\quad 4\mu_{03} \mu_{21}^3 - 3\mu_{21}^3 \mu_{12}^3) / \mu_{00}^{10} \\
AMI_3 &= (\mu_{20}(\mu_{21} \mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30} \mu_{03} - \mu_{21} \mu_{12}) + \\
&\quad \mu_{02}(\mu_{30} \mu_{12} - \mu_{21}^2)) / \mu_{00}^7
\end{aligned}$$

All these features are sufficient to characterize the shape of an image. The rationale behind the choice of these moments is that we are interesting in translation, rotation, scale, and projective transform invariance in order that the location, orientation, and scaling of the shape do not affect the extracted features. Further information on these approaches is discussed in [11].

4 Fuzzy Clustering for the Image Arrangement

The clustering algorithms achieve a partitioning of given data into clusters. In general a partition holds two properties: homogeneity within the clusters (data in a cluster must be similar) and homogeneity between clusters (isolation of a cluster from one another: data of different clusters have to be as different as possible).

The data are opportunely translated into a matrix, where each row is a characteristic vector which represents an image. In fact, the images set has been processed to pull out such data matrix, whose rows and columns are respectively the collected images and the relative extracted features. In this study, we are going to apply a fuzzy approach of clustering, the well-known *fuzzy C-Means* (briefly FCM) algorithm [5]. FCM represents the most common fuzzy clustering, particularly useful for flexible data organization. It takes as input a collection of patterns of a universe U (in our case, the collection of images) in form of matrix and produces fuzzy partitions of the given patterns (i.e. images) into (prefixed) c clusters.

The FCM algorithm recognizes spherical clouds of points (clusters) in a multi-dimensional data space and each cluster is represented by its center point (prototype). This process is completely unsupervised, aimed at identifying some inherent structures in a set of data.

The fuzzy version of clustering produces a more flexible partitioning of data. Precisely, each pattern (in our case, an image) is not associated exclusively to a cluster, but it can belong to more than one. After the fuzzy clustering execution, each pattern has associated a c -dimensional vector, where each cell represents the membership (in the range $[0, 1]$) of that pattern to each cluster.

Compared to the crisp version, the fuzzy clustering generates a flexible partitioning, more intuitive to interpret: a pattern can have some characteristics that are natively representative of more than one cluster, and the exclusive belonging to one cluster is a too restricted condition. In the fuzzy approach, the membership values better reveal the nature of data set and allow a clearer data analysis. Anyway, it is conceivable to assign a pattern to the cluster, whose membership is the highest.

More formally, each row of the matrix is a vector that represents an image $I \longleftrightarrow \underline{x} = (x_1, x_2, \dots, x_h)$, where each component of vector is a value computed

for a feature. The FCM algorithm aims at minimizing the objective function constituted by the weighted sum of the distances $dist_{i,k}$ between data points $\underline{x}_k = (x_{k,1}, x_{k,2}, \dots, x_{k,h})$ and the centers (or prototypes) $\underline{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,h})$, according to this formula:

$$Q(U, c) = \sum_{i=1}^c \sum_{k=1}^n u_{i,k}^m (dist_{i,k})^2 \tag{1}$$

where $c \geq 2$ is the number of clusters, $u_{i,k} \in [0,1]$ is the membership degree of \underline{x}_k ($k=1, \dots, n$) in the i -th cluster A_i ($i=1, \dots, c$), and $m > 1$ is the fuzzifier, which controls the quantity of fuzziness in the classification process (common choice of fuzzifier is $m = 2$) and finally $dist_{i,k}$ is:

$$dist_{i,k} = dist(\underline{x}_k, \underline{v}_i) = \sqrt{(\|\underline{x}_k - \underline{v}_i\|^2)} \tag{2}$$

just represents the euclidean distance between the data \underline{x}_k and the center \underline{v}_i of the i -th cluster.

In details, $U = (u_{i,k})$ is a $c \times n$ matrix of cluster memberships satisfying some constraints. In particular, M_{fc} is a family of fuzzy partition matrices:

$$M_{fc} = \left\{ U \mid u_{i,k} \in [0, 1]; \sum_{i=1}^c u_{i,j} = 1; 0 < \sum_{k=1}^n u_{i,j} < n, \forall i, j \right\}, \tag{3}$$

and $V = (\underline{v}_1, \dots, \underline{v}_c)$ is the ordered collections of cluster centers.

In our study, the data matrix is composed of n images, each one with h values, associated to the corresponding features. The FCM algorithm produces a partitioning of this collection into a prefixed number c of clusters.

The algorithm finds an optimal fuzzy partition of the data, which is carried out through an iterative optimization of **(1)**. Main steps are given as follows.

1. Choose the values c , m and a small positive constant ϵ ; then, generate randomly a fuzzy c -partition U^0 and set iteration number $t = 0$.
2. Given the membership values $u_{i,k}^{(t)}$, the cluster centers $v_i^{(t)}$, ($i = 1, \dots, c$) are calculated by

$$v_i^{(t)} = \frac{\sum_{k=1}^n (u_{i,k}^{(t)})^m x_k}{\sum_{k=1}^n (u_{i,k}^{(t)})^m} \tag{4}$$

3. Given the new centers $v_i^{(t)}$, update the membership value $u_{i,k}^{(t)}$:

$$u_{i,k}^{(t+1)} = \frac{1}{\sum_{j=1}^c \left(\frac{dist_{i,k}^2}{dist_{j,k}^2} \right)^{\frac{1}{m-1}}} \tag{5}$$

4. The process stops when $|U^{(t+1)} - U^{(t)}| < \epsilon$, otherwise go to step 2.

Let us note the only actual parameter of this algorithm is the number c of clusters. In general, this number is not known a priori. Selecting a different number of initial clusters can effectively affect the final partitioning of the data. The problem for finding an optimal c is usually called cluster validity **[3]**. The objective is to find optimal c clusters that can validate the best description of the

data structure. Each of these optimal c clusters should be compact and separated from other clusters. In the literature, many heuristic criteria have been proposed for evaluating fuzzy partitions; some of traditional cluster validity indexes, which have been frequently used, are Bezdek's partition coefficient (PC) [4], partition entropy (PE) [3], Xie-Beni's index [31].

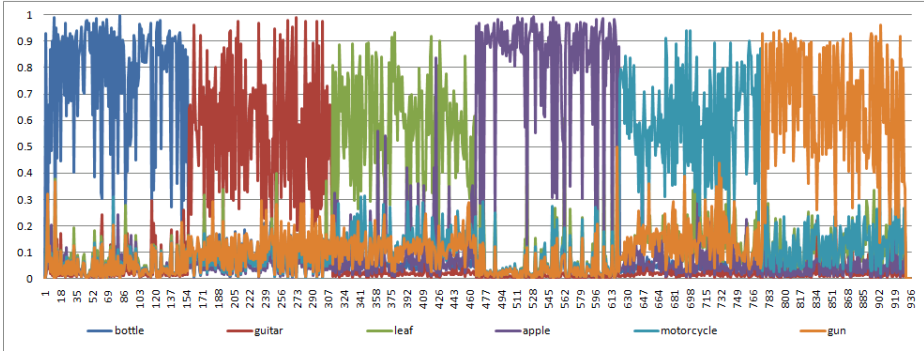


Fig. 1. The membership distribution among six clusters, produced by the FCM with $c = 6$ and $m = 2$

5 Experimental Results

The first experiment exploits a collection of images downloaded through Google images¹. The testbed consists of a sample of 930 images, composed of six classes of 155 images, ranked as follows: images in the range 1-155 represent bottles; in the range 156-310 there are images of guitars, then the leaves are in the range 311-465, the images of apples cover the range 466-620, the motorcycle images are in 521-775 and finally the last images set consist of guns in the range 776-930.

The test considers all the features presented above: geometrical features (E, R, C, π_{gen}), invariants moments features ($I_1, I_2, I_3, I_4, I_5, I_6, I_7$), and affine moments features (AMI_1, AMI_2, AMI_3). Then, the FCM algorithm has been executed considering the number of clusters equals to the number of images categories ($c = 6$). The final partitioning is sketched in Figure 1: each line represents the membership distribution in a cluster; in particular, each cluster is in correspondence with a class of images. For instance, the blue line in Figure 1 describes the memberships distribution of a cluster that represents to the class of bottles (first 155 data). Due to the fuzzy approach, the individual image membership can be distributed among all the clusters and assume a value in the range $[0,1]$ according to how it belongs to each cluster. The fuzzy method of clustering reveals more flexibility in the distribution of data: an image can belong to more than one cluster, because it shares similar characteristics with other

¹ The dataset can be downloaded at: <http://www.dmi.unisa.it/people/senatore/www/dati/dataset.rar>

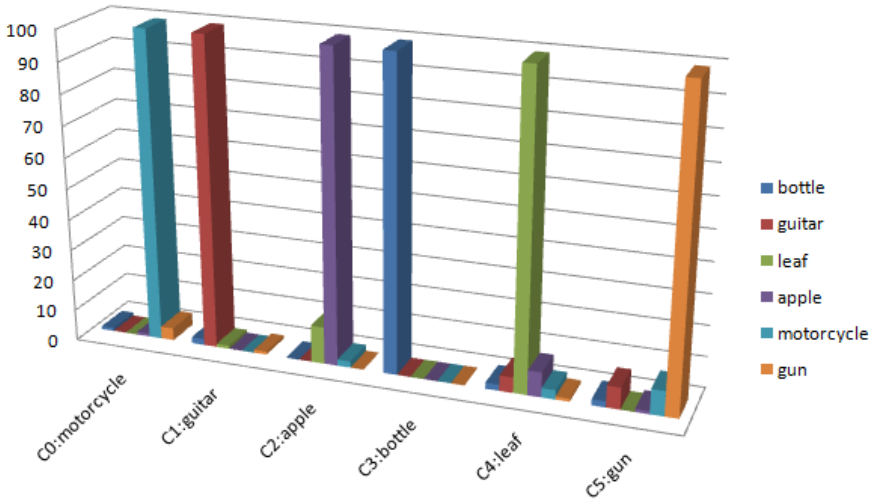


Fig. 2. The membership distribution among six clusters, associated to image classes

images, even though these latter belong to other clusters. It is licit to assume an image belongs to a given cluster, if its membership value for that cluster is the highest one. Figure 2 shows instead, a “synthetic” representation of this image distribution among the clusters, through histogram-based graphs. Each cluster represents a class of images. The clustering results are satisfying, because each class of images is almost completely individuated and associated to a cluster. In particular, in this specific testbed, classification error is quite restrained, as evidenced in Table 1, where the assessment of the clustering results is shown for each class/cluster. Each row provides the name of the class, the *misclassified* images, i.e. those images that have the highest membership in another class, different by the expected one, the *undecided* images, viz. all the images which membership is almost equally distributed among two or more clusters. Then the local *recall* and *precision* that is evaluated for each cluster.

More specifically, in the image retrieval context, the definition of recall and precision can be as follows:

$$Recall = \frac{\text{relevant retrieved images}}{\text{relevant images}} \quad (6)$$

$$Precision = \frac{\text{relevant retrieved images}}{\text{retrieved images}} \quad (7)$$

where the *relevant images* are the images which are expected in a certain class, the *retrieval images* are all the (correct and incorrect) images which are returned in that cluster, while the *relevant retrieved images* are just the images that really belong to the right cluster, associated to the correct class. Figure 1 reveals clusters associated to the leaves and motorcycle classes present the lowest membership distribution, even though the most of data are well placed in the cluster. In particular, let us analyze the class of leaves: most of misclassified data appear

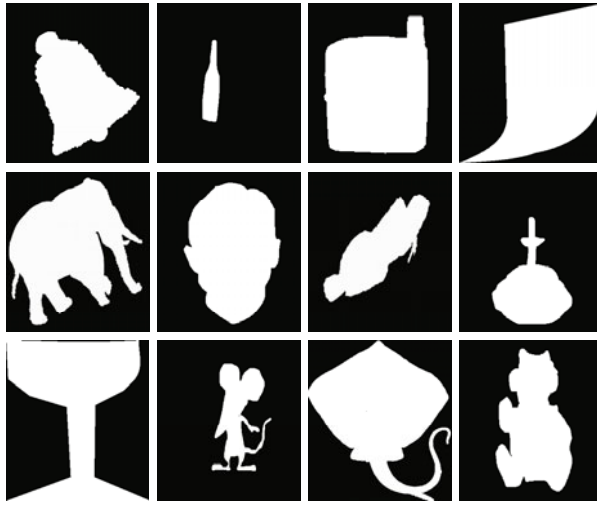


Fig. 3. Sample images representing classes of MPEG-7 CE data set used in the experiment



Fig. 4. Some samples used for the experiments. The entire dataset is composed of 930 images subdivided in six categories; bottles, leaves, guitars, motorcycles, guns, and apples.

in cluster of apples; this is due to the different shapes of leaves: after the image processing, some leaves present rounded shapes that can be easily confused with apples. In fact geometrical feature as P_1 and compactness assume values assimilable to those ones of apples. Indeed, the image numbered 424 for instance, is misclassified presenting highest membership value in the apple cluster: its distribution among cluster is [0.027 0.007 0.048 0.836 0.043 0.035], respectively for the clusters associated to the bottle, guitar, leaf, apple, motorcycle and gun classes. It is evident its highest membership value 0.836 in apple cluster versus 0.048 of the right cluster. Anyway, no image of leaves cluster is undecided. In the cluster of motorcycles, instead, two images are undecided: the numbers 746 and 772 with distribution membership [0.057 0.038 0.276 0.044 0.290 0.293] and [0.145 0.170 0.180 0.082 0.208 0.212] respectively. In fact, the highest membership values appear equally distributed among the clusters of motorcycles and guns.

The lowest membership distribution in the cluster of leaves yields worse precision values. The recall is computed on 142 well-classified relevant images, considering all the 155 image of the class. The precision, instead is evaluated as ratio between the 142 well-classified images and all the retrieved images in this class, i.e. 160 images among correct and incorrect ones. Similar considerations can be done for the cluster of guns: here, the retrieved images are 163 even though the well classified images are 149. The overall result emphasizes the efficacy of this approach: the experiment can be considered satisfactory, because presents well-defined classes, composed of most of relevant images.

The next experiment considers a subset of the MPEG-7 Core Experiment Shape-1 dataset, which is frequently used to evaluate shape matching and recognition algorithms. In particular, we have used the MPEG-7 CE Shape-1 Part-B dataset [11], composed of 70 shape categories, each of which has 20 samples with in-plane rotations, articulations, and oclusions. MPEG-7 CE Shape-1 Part-B data set includes 1400 shape samples, 20 for each class. We have used twelve shape classes, considering all the twenty shape samples. We have chosen following twelve classes: bell, bottle, cellular phone, comma, elephant, face, fish, fountain, glasses, rat, ray, teddy, shown in Figure 3. The shape classes are very distinct, but the data set shows substantial within-class variations.

The fuzzy clustering setting considers just 12 clusters, one for each class of the presented images set (totally 240 images) and exploits all the features defined in Section 3. In other words, a 240×14 input matrix is given as input to FCM

Table 1. Class-based evaluation of fuzzy clustering results

Classes	# Misclassified.	# Undecided.	Recall. %	Precision. %
bottle	5	2	95	100
guitar	12	0	92	97
leaf	13	0	91	88
apple	9	1	93	91
motorcycle	11	2	91	95
gun	6	0	96	91

Table 2. Confusion matrix relative to a subset of MPEG-7 CE Shape-1 Part-B dataset

		ACTUAL												
		bell	bottle	cellular phone	comma	elephant	face	fish	fountain	glass	rat	ray	teddy	
PREDICTED	bell	20	0	0	0	0	0	0	0	0	0	0	0	
	bottle	0	20	0	0	0	0	0	0	0	0	0	0	
	cellular phone	0	0	19	0	0	0	0	0	0	0	0	0	
	comma	0	0	0	15	0	0	0	0	0	7	0	0	
	elephant	0	0	0	0	17	0	0	0	0	2	0	0	
	face	0	0	0	0	0	20	0	0	0	0	0	0	
	fish	0	0	1	0	0	0	20	0	0	0	0	0	
	fountain	0	0	0	0	0	0	0	20	0	0	0	0	
	glass	0	0	0	0	0	0	0	0	15	0	0	0	
	rat	0	0	0	0	0	0	0	0	5	20	0	0	
	ray	0	0	0	5	3	0	0	0	0	0	11	0	
	teddy	0	0	0	0	0	0	0	0	0	0	0	20	

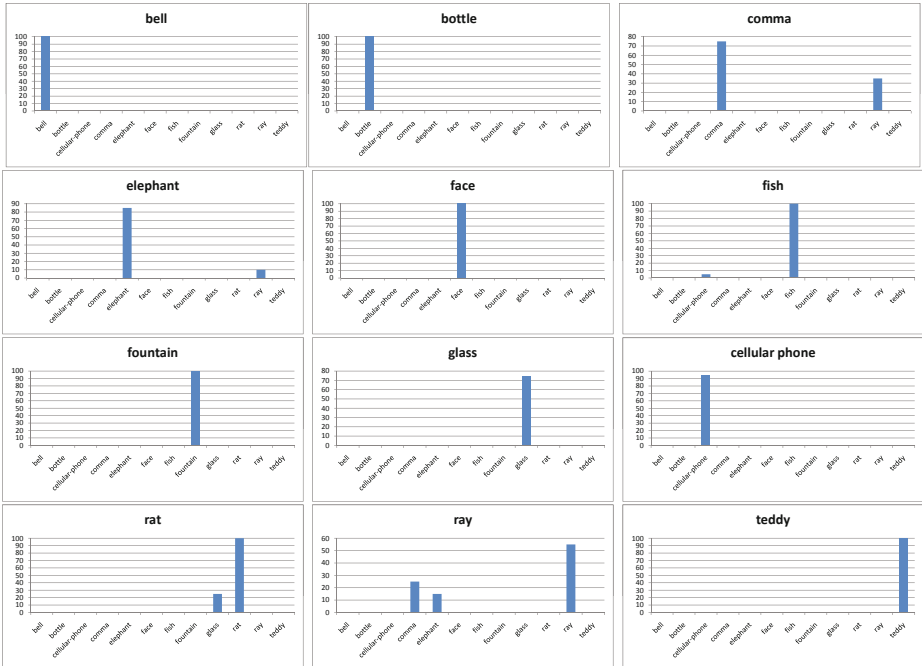


Fig. 5. Images distribution among the clusters produced by the FCM

algorithm. After the loading of these images and the image processing, the clustering phase has been started. Figure 5 shows the detailed classification results, where images belonging to different classes are allocated within the individual clusters. Table 2 synthesizes the results, showing the confusion matrix associated to this experiment. Let us note that many correspondences are revealed between the generated clusters (*predicted*) and the given (*actual*) images classes. In particular, some clusters look very homogeneous; most of them includes averagely about 80% of proper images. Just to give some example, the elements of the classes represented by fish, face, bottle, etc. appear all collocated in each individual cluster (100% of individuals are placed in each of them). This is not true any longer for the clusters concerning the rays and commas, even though we have a low overlap among categories. Finally, some clusters are representative of a specific class, even though elements of other class appear in them (in Figure 5 see the clusters representing the classes of rats, elephants, etc.).

6 Conclusion

The approach achieves an image classification and content-based retrieval. An initial image analysis allows the elicitation of visual features which are exploited to characterize the image through its shape. The fuzzy clustering techniques enable a relaxed distribution of images (compared to the crisp clustering); moreover they are robust respect to an image segmentation approaches based on k-means segmentation which meet some difficulties foreground and background colors do not contrast sufficiently. The effectiveness of this approach is evaluated through Information Retrieval measures, which reveals discrete performance.

This approach exploits a fuzzy clustering technique which, even though requires an a-priori fixed number of clusters, avoids overfitting of parameters and does not require a learning/training phase. In fact, our approach could be also considered non-parametric, if the only parameter, i.e., the number c of clusters is a-priori known. Otherwise, as said, methods based on cluster validity indexes [3], [31] find the optimal c and evaluate the fitness of partitions produced by clustering algorithms. Finally, the approach is robust respect to an image segmentation approach based on k-means segmentation which performs not very well when foreground and background colors do not contrast sufficiently.

Future extensions of this work foresee a development of a GUI-based application which supports the features extraction and the clustering technique. Additional features have been taken into account, particularly, some moments that are invariant to elastic transformations and convolution. We are going to extend the application, designing an visual query interface for the submission of a free hand drawing shape. This way, a ranked list of images whose shape is similar to the sketched one will be returned. Moreover, additional experiments with increased size and comparisons with other classification techniques have been taken into account.

References

1. Shape data for the mpeg-7 core experiment ce-shape-1, <http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz>
2. Aguilera, A., Suber, A., Martinez, L., Subero, A., Tineo, L.: Fuzzy image retrieval system. In: The 10th IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1247–1250 (2001)
3. Bezdek, J.: Cluster validity with fuzzy sets. *J. Cybernet.* 3, 58–72 (1974)
4. Bezdek, J.: Numerical taxonomy with fuzzy sets. *J. Math. Biol.* 1-1, 57–71 (1974)
5. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell (1981)
6. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: *CVPR 2008: Proceedings of the Conference on Computer Vision and Pattern Recognition* (2008)
7. Celebi, M.E., Aslandogan, Y.A.: A comparative study of three moment-based shape descriptors. In: *ITCC 2005: Proceedings of the International Conference on Information Technology: Coding and Computing*, pp. 788–793. IEEE Computer Society, Washington, DC, USA (2005)
8. Chen, Y., Wang, J.: A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(9), 1252–1267 (2002)
9. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys* 40(2), 1–60 (2008)
10. Flusser, J., Suk, T.: Pattern recognition by affine moment invariants. *Pattern Recognition* 26(1), 167–174 (1993)
11. Flusser, J., Zitova, B., Suk, T.: *Moments and Moment Invariants in Pattern Recognition*. Wiley, Chichester (2009)
12. Frigui, H.: Membershipmap: Data transformation based on granulation and fuzzy membership aggregation. *IEEE Transactions on Fuzzy Systems* 14(6), 885–896 (2006)
13. Gao, Y., Fan, J.: Semantic image classification with hierarchical feature subset selection. In: *MIR 2005: Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 135–142. ACM, New York (2005)
14. Gevers, T., Smeulders, A.W.M.: Image search engines - an overview. In: Medioni, G., Kang, S.B. (eds.) *Emerging Topics in Computer Vision*. Prentice-Hall, Englewood Cliffs (2004)
15. Ha, J.-Y., Kim, G.-Y., Choi, H.-I.: The content-based image retrieval method using multiple features. In: *NCM 2008: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 652–657. IEEE Computer Society, Washington, DC, USA (2008)
16. Hall, L., Bensaid, A., Clarke, L., Velthuizen, R., Silbiger, M., Bezdek, J.: A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Transactions on Neural Networks* 3(5), 672–682 (1992)
17. Hu, M.-K.: Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* 8(2), 179–187 (1962)
18. Images, Z.C., Zhang, M., Hall, L.O., Muller-karger, F.E., Goldgof, D.B.: Knowledge-guided classification of coastal. *International Journal of Pattern Recognition and Artificial Intelligence* 14, 987–1007 (2000)

19. Jingwei, S.E., Eschrich, S., Ke, J., Hall, L.O., Goldgof, D.B.: Fast fuzzy clustering of infrared images. In: Joint 9th IFSA World Congress and 20th NAFIPS International Conference (2001)
20. Krishnapuram, R., Medasani, S., Jung, S.-H., Choi, Y.-S., Balasubramaniam, R.: Content-based image retrieval based on a fuzzy approach. *IEEE Transactions on Knowledge and Data Engineering* 16(10), 1185–1199 (2004)
21. Kulkarni, S., Verma, B., Sharma, P., Selvaraj, H.: Content based image retrieval using a neuro-fuzzy technique, vol. 6, pp. 4304–4308 (July 1999)
22. Lay, J.A., Guan, L.: Image retrieval based on energy histograms of the low frequency dct coefficients. In: ICASSP 1999: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3009–3012. IEEE Computer Society, Washington, DC, USA (1999)
23. Lieberman, H., Rozenweig, E., Singh, P.: Aria: An agent for annotating and retrieving images. *Computer* 34, 57–62 (2001)
24. Lowe, D.G.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (August 1999)
25. Nachttegael, M., der Weken, D.V., Witte, V.D., Schulte, S., Mélange, T., Kerre, E.E.: Color image retrieval using fuzzy similarity measures and fuzzy partitions. In: *ICIP* (6), pp. 13–16 (2007)
26. Pengyu, L., Kebin, J., Peizhen, Z.: An effective method of image retrieval based on modified fuzzy c -means clustering scheme, vol. 3 (November 2006)
27. Rath, T.M., Manmatha, R., Lavrenko, V.: A search engine for historical manuscript images. In: *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 369–376. ACM, New York (2004)
28. Rui, Y., Huang, T.S., Chang, S.-F.: Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation* 10(1), 39–62 (1999)
29. Tian, Y., Mei, L.: Image retrieval based on multiple features using wavelet. In: *ICCIMA 2003: Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, p. 137. IEEE Computer Society, Washington, DC, USA (2003)
30. Vertan, C., Boujemaa, N.: Embedding fuzzy logic in content based image retrieval, pp. 85–89 (2000)
31. Xie, X., Beni, G.: A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 841–847 (1991)
32. Yao, W., Hall, L., Goldgof, D., Muller-Karger, F.: Finding green river in seawifs satellite images. In: *International Conference on Pattern Recognition*, vol. 2, p. 2307 (2000)
33. Zhang, D.: Improving image retrieval performance by using both color and texture features. In: *ICIG 2004: Proceedings of the Third International Conference on Image and Graphics*, pp. 172–175. IEEE Computer Society, Washington, DC, USA (2004)
34. Zhang, D., Lu, G.: Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Syst.* 9(1), 15–30 (2003)

Quantum Logic Based MPEG Query Format Algebra

Mario Döller¹, Sebastian Lehrack², Harald Kosch¹, and Ingo Schmitt²

¹ Department of Distributed Information Technology
University of Passau
Passau, Germany

² Department of Computer Science
Brandenburg University of Technology
Cottbus, Germany

Abstract. The need for fast processing of query requests in multimedia retrieval systems is apparent. One basis for optimization is the formalization of the corresponding query language by a respective algebra. Furthermore, an algebra is important for demonstrating the profoundness and validity of a query language. In this context, the article contributes a formal semantics model for the novel standardized MPEG Query Format for multimedia search. In addition to the specification of its syntax and semantics, our quantum logic approach for fuzzy retrieval on behalf of the formal model is discussed. Besides the validity of our formalization is demonstrated on some examples, the advantages as well as the shortcomings of the query format are discussed.

Keywords: Multimedia retrieval, MPEG Query Format, Fuzzy Retrieval, Query Language Algebra.

1 Introduction

Multimedia queries unify based on their retrieval characteristic the worlds of two evaluation logics, namely similarity or fuzzy retrieval and exact retrieval on concrete data. For instance the request *Give me all images where the file size is 50 KB and whose color histogram is similar to the given one* contains a true/false evaluation on the image's file size and an imprecise evaluation between color distributions. For enhanced requirements this example could also be improved by factoring weighting functionality (e.g., color similarity is more important than the file size constraint) or taking temporal and spatial conditions into account.

Related to the example request, there have been developed a numerous amount of multimedia query languages and systems that tackle subareas of multimedia request types. For instance, some are specialized to temporal [16] or spatial [14] requests. In addition, as metadata of multimedia is very often expressed in XML instances, XML repositories and XML query languages such as XQuery (and their derivatives for multimedia e.g. [24]) can also be used for multimedia retrieval. Although, there is a large diversity of individual solutions, a universal query language supporting most of the requirements for multimedia retrieval is still missing.

Due to this fact, the ISO/IEC consortium in its SC29WG11 (MPEG) subgroup standardized the *MPEG Query Format* (MPQF) [7] which covers most multimedia search scenarios. The origin of fast processing of query requests lies in a formal model of the underlying query language and its optimization capabilities. In this context, the main contributions of this paper can be summarized as follows: First, a formal model of the newly standardized MPEG Query Format is introduced. This formal model provides a sound theoretical foundation for processing MPQF requests in multimedia database systems. Here especially, the abstract concepts on the underlying data model, the syntax of involved operations and its semantics during the evaluation are described. Besides, the use of a novel quantum logic approach for fuzzy retrieval on behalf of the formal model for MPQF is discussed. Finally, this paper highlights advantages as well as shortcomings of the novel standard.

The remainder of this paper is organized as follows: Section 2 introduces related work in the area of XML based and multimedia based algebras. This is followed by an overall description of the MPEG Query Format and its data model in Section 3. A formal syntactical and semantical specification of an algebra for the MPEG Query Format is stated in Section 4. In Section 5 two evaluation models (fuzzy logic, quantum logic) are applied to the algebra and an example evaluation is demonstrated. Finally, the article is concluded in Section 6. Note, due to space concerns this article contains an excerpt of the full syntactical and semantical specification of the developed algebra. The full specification can be obtained in the technical report at: http://dimis.fim.uni-passau.de/iris/mpqf_algebra_TR.pdf

2 Related Work

Based on the fact that the MPEG Query Format tackles both worlds (XML and multimedia retrieval), this section highlights related work describing existing algebra for XML based and multimedia query languages, respectively.

2.1 Algebra for XML Based Query Languages

In general, many metadata formats use XML scheme (e.g. MPEG-7¹, TV-Anytime²) for representing annotations of multimedia content. Consequently, query languages that address XML data can be applied in a restricted way for answering multimedia requests. In this context, available algebra for those languages can be categorized, according to tuple based and tree based approaches. Representatives of tuple based algebras are for instance Natix algebra [4] or BSA algebra [19]. All these algebra derive the well known tuple based approach of the relational world of databases where the nodes in an XML document are mapped to tuples and its child nodes are the respective attributes. For instance the algebra defined in [4] extends the semantic of the relational algebra by means of

¹ http://mpeg.chiariglione.org/working_documents.htm#MPEG-7

² <http://www.tv-anytime.org/>

enhanced data types within the XPath data model (XDM)³ and the redefinition of *Select*, *Project* and *Join* operations. In addition, operations for the navigation and addressing of nodes and values (e.g., *getValue*) are defined. The evaluation executes the given filter predicates on tuple sequences containing location steps (describes navigation steps in the document tree).

Algebra of the second approach focus on a tree based representation (e.g., TAX [9], TLC [17]). For instance, in the data model of [9], a set of trees (*Data-Tree*) form the basis of the representation. During the evaluation, the XML document is partitioned in *Data-Trees* and matched against a *Pattern-Tree* identifying the filter criteria. During this pattern matching process so called *Witness-Trees* are extracted that form candidates for the result set.

2.2 Algebra for Multimedia Query Languages

Besides algebras for XML processing, algebras especially designed for multimedia data have been emerged in the literature. One of the first approaches in this area has been introduced in [1]. Here, the authors define based on similarity measures over objects with some properties (e.g., color feature) a Multimedia Similarity Algebra (*MSA*) and similarity algebra operations (e.g., Sim-Union Join). For evaluation, this algebra has been converted into a relational model (*rMSA*) and implemented on top of the I.SEE (Integrated SEArch Engine) system. By improving some shortcomings of the *MSA* approach (e.g., no combination of similarity and relational operators was foreseen), the authors in [2] proposed a similarity based algebra for multimedia databases especially for OR database models. In this context, a multimedia join approach has been detailed by the same authors in [11]. Similar to the *MSA* algebra [1], the *SAME^W* algebra [5] focus on imprecision and user preferences in multimedia queries. For this purpose, the relational algebra concepts have been extended by generic scoring functions for logical operators (AND, OR).

Recent, in [23] a similarity algebra (*SA*) has been proposed featuring weighted similarity predicates and a formally founded derivation of a similarity relational calculus. In contrast to the *SAME^W* algebra, the semantic of the introduced weighting operators is open and special care has been aligned to the side effects of weighted conjunctions. An extension of the relational model for injecting imprecision and uncertainty has been applied in [3] by integrating ranked tables over domains with similarity. A novel approach based on quantum logic has been introduced in [21]. Here the advantages of mixing classical conditions with proximity retrieval conditions lie in embodying the whole underlying query semantic. Furthermore, it has been shown that quantum conjunction, disjunction and negation conform the rules of probability theory.

3 The MPEG Query Format

The MPEG Query Format (MPQF) [7] standard specifies a format for the interaction of multimedia clients and multimedia retrieval systems (MMRS). In

³ <http://www.w3.org/TR/xpath-datamodel/>

detail, the standard defines the message format for multimedia requests (e.g. Query by Example or Query By Text) to heterogeneous MMRS and the message format for their responses. Furthermore, a management part provides features such as service discovery (service is a synonym for MMRS) and service capability description.

MPQF came from the MPEG-7 activities but it is important to note that MPQF is not tied to MPEG-7 [15] at all. In fact any XML based metadata format can be integrated. The interested reader is referred to [7] for detailed information of the MPEG Query Format.

3.1 The MPEG Query Format Data Model

The MPEG Query Format is an XML based multimedia retrieval language addressing data that is stored in XML instance documents. A subset of MPQF bases on XPath 2.0 and XQuery 1.0 which implies the use of the XQuery and XPath Data Model (XDM). The XDM defines the data types and concepts valid for the XQuery, XPath and XSLT⁴ languages. It bases on the *Infoset*⁵ (contains definitions for the information in a well-formed XML document) and extends it among others by typed atomic values and for ordered heterogeneous sequences. Instances of the data model are organized as ordered flat sequences (e.g., a sequence is not allowed to contain other sequences) of items. In series, an item is either a node (e.g., element, attribute) or an atomic value.

In order to fulfill the additional requirements of MPQF (e.g., new data types, scoring including user preferences and thresholds, additional evaluation logic) the XDM model needs to be enhanced to keep the *closure* characteristic of our query language.

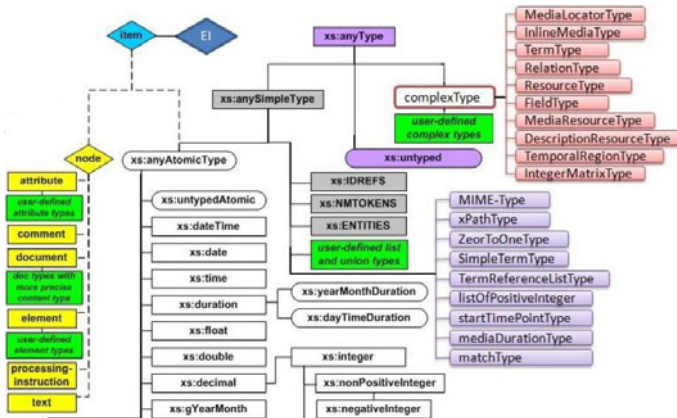


Fig. 1. XDM extended MPQF data model

⁴ <http://www.w3.org/TR/xslt>

⁵ <http://www.w3.org/TR/xml-infoset/>

In this context, XDM is extended by the following means (see Figure 1 which shows an excerpt of the full XDM including its MPQF extension). First of all, the list of simple types has been enhanced by extending the *anySimpleType* type. This reflects the need to extend XDM at the same level as atomic values. For instance, a new *zeroToOneType* type is introduced to fix the internal score values to the $[0, 1]$ interval. Similar extensions are needed in the domain of complex types (note that a new subtype has been introduced). Here, the set of types is extended by means to describe complex multimedia descriptions (*MediaResourceType*, *DescriptionResourceType*) or by specifying spatial or temporal relations (*RelationType*).

Most of the definitions given in XDM are also valid for the extended version. However, the following enhancements need to be considered. Similar to XDM, the processing in MPQF operates on sequences of an ordered collection of *evaluation items (EI)* (derived by the original item construct of XDM). An EI is an element of the extended XDM, namely a node an atomic value or one of the introduced complex types. In this context, the scope of query evaluation and the granularity of the result set can be determined by an *EvaluationPath* element specified within the query. If this *EvaluationPath* element is not specified, the output result is provided as a collection of multimedia content, as stored in the repository, all satisfying the query condition.

4 Algebra for MPEG Query Format

An MPQF input query may consist of the following seven elements and attributes, namely *QFDeclaration*, *OutputDescription*, *QueryCondition*, *Service-Selection*, *previousAnswerID*, *immediateResponse* and *Timeout*. It is assumed that a pre-processing step takes care of most of the mentioned elements and attributes. The algebra only covers the *OutputDescription* and *QueryCondition* part and translates those elements by using defined operations into an algebra expression for further processing.

The MPQF algebra operates on a tuple set which extends the relational algebra for fuzzy multimedia retrieval capabilities including preference and threshold values. The following subsections introduces syntax and semantics of the MPQF algebra. Moreover, we give an impression about mapping rules transforming MPQF queries into MPQF algebra expressions.

4.1 Syntax

Definition 1: The MPQF algebra is denoted as a tuple of the form $(\mathbf{U}, \mathbf{C}, \mathbf{D}, \mathbf{Dom}, \mathbf{R}, \Theta, \mathbf{T}, \mathbf{F}, \Delta)$ which is defined as follows: \mathbf{U} is the universe of all attributes, \mathbf{C} is the set of all constants, \mathbf{D} is the set representing the domain of all types, \mathbf{Dom} is the mapping of all attributes, functions and operations to a specific domain, \mathbf{R} is a finite set of relation schemes (R_1, R_2, \dots, R_n) , where each relation schema is a subset of \mathbf{U} , Θ is a set of weighting variables, \mathbf{T} is a set of threshold variables, \mathbf{F} is a set of all functions, Δ is the set of

predicates $\Delta = \Delta_{SP} \cup \Delta_{MP}$ whereby Δ_{SP} covers simple Boolean predicates $\Delta_{SP} = \{=, <, >, \leq, \geq, \neq\}$ and Δ_{MP} contains MPQF specific multimedia predicates. The specific MPQF multimedia predicates are defined as follows (full specification in the technical report):

- **QueryByMedia:** $QBM_{exact}(\text{targetResource}, \text{testResource})$ and $QBM_{similar}(\text{targetResource}, \text{testResource}, \tau)$ whereby $\text{Dom}(\text{targetResource}) = \text{Dom}(\text{testResource}) = \text{MediaResourceType}$ and $\tau \in \mathbf{T}$ is a threshold.
- **QueryByDescription:** $QBD_{exact}(\dots)$ and $QBD_{similar}(\dots)$,
- **QueryByFreeText:** $QBFT_{freeText}(\dots)$ and $QBFT_{regularExp}(\dots)$,
- ...

Moreover, specific MPQF predicates are classified in exact and similarity predicates $\Delta_{MP} = \Delta_{MP_{exact}} \cup \Delta_{MP_{similar}}$:

- $\Delta_{MP_{exact}} = \{QBM_{exact}, QBD_{exact}, QBFT_{regularExp}, QBFR_{range}, QBFR_{dist}, SQ, TQ, QBXQ, QBROI_{exact}^{temporal}, QBROI_{exact}^{spatial}\}$
- $\Delta_{MP_{similar}} = \{QBM_{similar}, QBD_{similar}, QBFT_{freeText}, QBRF, QBROI_{similar}^{temporal}, QBROI_{similar}^{spatial}\}$

Definition 2: An expression E in the MPQF algebra is recursively defined as follows (E_1, E_2 are MPQF algebra expressions and $\text{attr}(E_1)$ gives all attributes of the relation scheme of E_1):

1. Relation: $E = R$, whereby $R \in \mathbf{R}$,
2. Projection: $E = \pi_{A_{p_1}, \dots, A_{p_n}}(E_1)$ where A_{p_i} stands for an attribute of $\text{attr}(E_1)$,
3. Selection: $E = \sigma_F(E_1)$ whereby F denotes a *logical formula* ($\text{attr}(F) \subseteq \text{attr}(E_1)$) defined over *terms*. Precisely, a *term* is defined as
 - a constant: if $c \in \mathbf{C}$ then c is a term, or
 - an attribute: if A_{p_i} is an attribute of $\text{attr}(E_1)$ then A_{p_i} is a term, or
 - a function over terms: if $f \in \mathbf{F}$ and t_1, \dots, t_n are terms then $f(t_1, \dots, t_n)$ is a term whereby $f(t_1, \dots, t_n)$ is correct typed.

Then, a *logical formula*

- is specified as a predicate over terms: if $\delta \in \Delta$ and t_1, \dots, t_n are terms then $\delta(t_1, \dots, t_n)$ is a formula, whereby $\delta(t_1, \dots, t_n)$ is correct typed, or
 - is constructed as $F_1 \wedge F_2, F_1 \vee F_2, F_1 \text{ XOR } F_2$ or $\neg F_1$ whereby F_1 and F_2 are logical formulas.
4. Join: $E = E_1 \bowtie_F E_2$, whereby F is a logical formula (see above) defined over attributes of $\text{attr}(E_1) \cup \text{attr}(E_2)$.

4.2 Semantics

Constitutively on the syntax of all components within the MPQF algebra, in a next step the semantics of MPQF expressions are introduced. For this purpose, we declare an interpretation over a MPQF algebra as:

Definition 3: An interpretation over a MPQF algebra syntax ($\mathbf{U}, \mathbf{C}, \mathbf{D}, \mathbf{Dom}, \mathbf{R}, \mathbf{\Theta}, \mathbf{T}, \mathbf{F}, \mathbf{\Delta}$) is a triple $\mathbf{d}, \mathbf{db}, \mathbf{I}$ where

- \mathbf{d} is a finite set of domains $\{d_1, d_2, \dots, d_n\}$, each domain is a non-empty set of values and \mathbf{db} is a finite set of finite relations $\{r_1, r_2, \dots, r_p\}$ over these domains.
- I is an interpretation function which defines domains, relation schemes, attribute domains, constants, weighting variables, thresholds and functions. Furthermore, the interpretation function I
 - maps any Boolean predicate and any exact multimedia predicate to a binary function: $\delta \in \Delta_{SP} \cup \Delta_{MP_{exact}} : I(\delta) : I(Dom(\delta)) \times I(Dom(\delta)) \rightarrow \{0, 1\}$,
 - maps any similarity multimedia predicate to a scoring function: $\delta \in \Delta_{MP_{similar}} : I(\delta) : I(Dom(\delta)) \times I(Dom(\delta)) \rightarrow [0, 1]$ and
 - maps any logical operator (\wedge, \vee, \neg) to a scoring function which depends on the underlying evaluation model (see Section 5).

Definition 4: The semantics of a MPQF algebra expression E are inductively defined by the interpretation function I^* :

1. **Relation:** $E = R \in R : I(R) = (1, v_0, \dots, v_n)(v_1, \dots, v_n) \in R$ where v_1, \dots, v_n are values of a tuple in R . Thus, all tuples are additionally equipped with an initial score value of 1 since they are considered as true facts.
2. **Projection:** $E = \pi_{A_{p_1}, \dots, A_{p_n}}(E_1)$: Let $sv_{0_1}, \dots, sv_{0_k}$ denote all score values for a fixed value list v_{p_1}, \dots, v_{p_n} where $(sv_{0_i}, v_1, \dots, v_k) \in I^*(E_1)$ holds and the corresponding values are identical: $p_i = j \Rightarrow v_{p_i} = v_j$ for $i = 1, \dots, n$. Then, $I^*(\pi_{A_{p_1}, \dots, A_{p_n}}(E_1)) = \{(sv, v_{p_1}, \dots, v_{p_n}) \mid (sv_{0_i}, v_1, \dots, v_m) \in I^*(E_1)\}$ where $sv = I(\vee)(sv_{0_1}, \dots, sv_{0_k})$.
3. **Selection:** $E = \sigma_F(E_1) : I^*(\sigma_F(E_1)) = \{(sv, v_1, \dots, v_m) \mid (sv_{old}, v_1, \dots, v_m) \in I^*(E_1) \wedge sv = I(\wedge)(sv_{old}, eval((v_1, \dots, v_m), F))\}$ whereby the logical formula F is evaluated by
 - $eval(t_i, F) = I(c)$ if F is a constant $c \in \mathbf{C}$,
 - $eval(t_i, F) = v_{p_j}$ if F is an attribute A_{p_j} ,
 - $eval(t_i, F) = I(f)(I(t_1), \dots, I(t_n))$ if F is a function $f \in \mathbf{F}$ over terms,
 - $eval(t_i, F) = I(\delta)(I(t_1), \dots, I(t_n))$ if F is a predicate $\delta \in \Delta$ over terms,
 - $eval(t_i, F_1 \wedge F_2) = I(\wedge)(eval(t_i, F_1), eval(t_i, F_2))$,
 - $eval(t_i, F_1 \vee F_2) = I(\vee)(eval(t_i, F_1), eval(t_i, F_2))$,
 - $eval(t_i, F_1 \text{ XOR } F_2) = I(\text{XOR})(eval(t_i, F_1), eval(t_i, F_2))$ and
 - $eval(t_i, \neg F_1) = I(\neg)(eval(t_i, F_1))$.
4. **Join:** $E = (E_1 \bowtie_F E_2) : I^*(\sigma_F(E_1)) = \{(sv, v_1, \dots, v_n, w_1, \dots, w_m) \mid (sv_{old_1}, v_1, \dots, v_n) \in I^*(E_1) \wedge (sv_{old_2}, w_1, \dots, w_m) \in I^*(E_2) \wedge sv = I(\wedge)(sv_{old_1}, sv_{old_2}, eval((v_1, \dots, v_n, w_1, \dots, w_m), F))\}$ whereby the formula evaluation $eval((v_1, \dots, v_n, w_1, \dots, w_m), F)$ is defined as above.

4.3 Mapping an MPQF Query to an MPQF Algebra Expression

In this section we sketch mapping rules which transform an MPQF query into an MPQF algebra expression. Particularly, we discuss the construction of an algebra expression including a selection or join condition with preference values.

QueryCondition. First of all, the QueryCondition section of an MPQF query determines the structure of the generated MPQF algebra expression. In detail, following mappings are possible:

- EvaluationPath \mapsto generate relation R by extracting all relevant attributes from the queried XML source,
- Condition $\mapsto \sigma_F(R)$ whereby R is determined by the corresponding EvaluationPath and F is constructed as BooleanExpressionType (see ExpressionType below) and
- JoinType $\mapsto \sigma_{F_1}(R_1) \bowtie_{F_2} \sigma_{F_3}(R_2)$ whereby the join condition F_2 is defined as BooleanExpressionType JoinCondition and $\sigma_{F_i}(R_i)$ is constructed as Condition (see above).

ExpressionType. By using instances of ExpressionType we can build a condition which is associated with a selection or a join operation. Basically, we differ logical expressions (BooleanExpressionType) to generate logical formulas from arithmetic and string expressions to construct terms:

- BooleanExpressionType:
 - ComparisonExpressionType \mapsto operation of $\{=, <, >, \leq, \geq, \neq\}$
 - QueryType (see corresponding subsection below)
 - AND, OR, NOT, XOR $\mapsto \wedge, \vee, \neg, \text{XOR}$
 - preferenceValue (see next subsection)
- ArithmeticExpressionType: Add, Subtract, ... \mapsto function of \mathbf{F}
- StringExpressionType: UpperCase, LowerCase, ... \mapsto function of \mathbf{F}

Integration Preference Values Into an MPQF Algebra Condition. The integration of preference values into an MPQF algebra condition is surprisingly simple, in contrast to the approach from Fagin and Wimmers [8]. At first we assign a weighting variable $\theta_i \in [0, 1]$ to each operand (subcondition) of a conjunction or disjunction, e.g. $c_1 \wedge_{(\theta_1, \theta_2)} c_2$. This weighting variable θ_i is derived from the XML field preferenceValue of the corresponding MPQF subcondition. It controls the influence of the score value produced by evaluating the subcondition c_i . The main idea of our weighting approach is the application of two syntactical substitution rules. They convert a weighted conjunction and a weighted disjunction into unweighted versions of the respective operations. For this purpose, we insert weighting constants as fixed score values into a MPQF algebra condition:

$$c_1 \wedge_{(\theta_1, \theta_2)} c_2 \rightsquigarrow (c_1 \vee \neg\theta_1) \wedge (c_2 \vee \neg\theta_2) \quad (1)$$

$$c_1 \vee_{(\theta_1, \theta_2)} c_2 \rightsquigarrow (c_1 \wedge \theta_1) \vee (c_2 \wedge \theta_2) \quad (2)$$

To elucidate the mechanism behind the substituted formulas we will examine two extreme cases in more detail. A weighting variable of 0 ($\theta_i = 0$) leads to a behaviour that the corresponding subcondition c_i has no longer any effect on the final evaluation result. On contrary, if both weight variables are equal to 1 ($\theta_1 = \theta_2 = 1$), we achieve the same evaluation result generated by applying the unweighted versions of conjunction and disjunction. For more details we recommend [20].

QueryType. Specific MPQF predicates as QueryByMedia or QueryByDescription are instances of QueryType. For example, MPQF predicates of type QueryByMedia are mapped by

- QueryByMedia \mapsto $\text{QBM}_{exact}(\text{targetResource}, \text{testResource})$ or $\text{QBM}_{similar}(\text{targetResource}, \text{testResource}, \tau)$ whereby
 - the value of matchType determines QBM_{exact} or $\text{QBM}_{similar}$,
 - the function getThis() returns a reference \mapsto targetResource and
 - MediaResource or MediaResourceREF \mapsto testResource.

OutputDescription. The elements of a OutputDescription specify a final projection and/or grouping operation:

- OutputDescriptionType $\mapsto \pi_{AttrList}(\gamma_{GAttrList, GAggFuncList}(E))$ whereby
 - $AttrList := (A_{p_1}, \dots, A_{p_n})$ is defined by ReqField $\mapsto A_{p_i}$,
 - $GAttrList := (A_{g_1}, \dots, A_{g_m})$ is defined by GroupByField $\mapsto A_{g_i}$,
 - $GAggFuncList := (AggFunc_{l_1}, \dots, AggFunc_{l_q})$ is defined by Aggregate $\mapsto AggFunc_{l_i}$ and
 - E is constructed by a QueryCondition (see above).

5 Evaluation Models

The semantics for a selection or join condition are an essential part of defining the evaluation of a MPQF algebra expression. Thus, we discuss fuzzy and quantum logic as two alternatives for the interpretation of logical operators $I(\wedge)$, $I(\vee)$ and $I(\neg)$ (see Section 4.2) in the following subsections. Especially, we emphasize the advantages of the quantum logic based evaluation against the fuzzy logic approach.

5.1 Fuzzy Logic

The main principle of fuzzy set theory is to generalise the concept of set membership [25]. In classical set theory a characteristic function $1_A : \Omega \rightarrow \{0, 1\}$ defines the memberships of objects $\omega \in \Omega$ to a set $A \subset \Omega$, whereby $1_A(\omega) = 1$, if $\omega \in A$ and $1_A(\omega) = 0$ otherwise. In fuzzy set theory the characteristic function is replaced by a membership function $\mu_M : \Omega \rightarrow [0, 1]$, that assigns numbers to objects $\omega \in \Omega$ according to their membership degree to a fuzzy set M . Membership degrees can be used to represent different kinds of imperfect knowledge, including *similarity*, *preference*, and *uncertainty*.

Conjunctions and disjunctions of fuzzy membership degrees are evaluated by special classes of functions called *t-norms* and *t-conorms*, respectively. For input values from $\{0, 1\}$, all *t-norms* and *t-conorms* behave like the Boolean conjunction and disjunction. For the values in between, however, different behaviours are possible. Zadeh in [25] suggests following evaluation functions:

$$eval(t, c) = \mu_c(t) \quad \text{if } c \text{ is atomic,} \tag{3}$$

$$eval(t, c_1 \wedge c_2) = \min(eval(t, c_1), eval(t, c_2)) \tag{4}$$

$$eval(t, c_1 \vee c_2) = \max(eval(t, c_1), eval(t, c_2)) \tag{5}$$

$$eval(t, \neg c) = 1 - eval(t, c) \tag{6}$$

whereby c_1 and c_2 are arbitrary subconditions. In this context, a good evaluation of fuzzy classifiers has been presented in [10].

5.2 Quantum Logic

In general, quantum logic enables the logic based construction of conditions starting from traditional Boolean and similarity predicates. The underlying idea is to apply the theory of vector spaces, also known from quantum mechanics and quantum logic, for query processing.

All attribute values of a tuple t are embodied by the direction of a normalised vector. The condition c itself corresponds to a vector subspace also called *condition space*. The evaluation result is then determined by the minimal angle between tuple vector and condition space. The squared cosine of this angle is a value out of the interval $[0, 1]$ and can therefore be interpreted as a similarity measure as well as a score value. A method for a convenient computation of the desired squared cosine of this angle is developed in [21]. It allows to evaluate a tuple t against a *normalised* (see below) condition c constructed by \wedge, \vee and \neg recursively as follows:

$$eval(t, c) = \varphi(t, c) \quad \text{if } c \text{ is atomic,} \tag{7}$$

$$eval(t, c_1 \wedge c_2) = eval(t, c_1) * eval(t, c_2) \tag{8}$$

$$eval(t, c_1 \vee c_2) = eval(t, c_1) + eval(t, c_2) - eval(t, c_1 \wedge c_2) \tag{9}$$

$$eval(t, \neg c) = 1 - eval(t, c) \tag{10}$$

whereby c_1 and c_2 are arbitrary subconditions. The function $\varphi(t, c)$ returns the evaluation of a single similarity predicate c . Its structure depends on the domain of the queried attribute of t . In general, any set of similarity values which can be produced by the scalar product of normalised vectors is supported. That is, the similarity values must form a semi-positive definite correlation matrix.

The defined operations [8] and [9] can only be applied, if the considered condition c is evaluated in a specific *syntactical form*. In this normal form only *mutually exclusive* subconditions or subconditions with *disjoint* sets of restricted attributes are allowed. The algorithm `norm` [21] transforms an arbitrary condition into the required normal form by using logical transformation rules as idempotence [6], absorption [7] and distributivity [8]. To preserve these logic laws we need following

⁶ Idempotence: $A \wedge A \equiv A$ and $A \vee A \equiv A$
⁷ Absorption: $A \vee (A \wedge B) \equiv A$ and $A \wedge (A \vee B) \equiv A$
⁸ Distributivity: $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ and $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

restriction: *In a valid condition any attribute must not be queried by more than one constant in a similarity predicate.* This restriction is respected by a MPQF algebra condition, if the quantum logic model is used for query processing.

5.3 Comparison Fuzzy and Quantum Logic

The functions *min/max* are the standard *t*-norm/*t*-conorm in fuzzy logic because it is the only idempotent² and first proposed set of functions [25]. Nevertheless, [12] shows that the application of *min/max* differs from the intuitional understanding of a combination of values, because the binary *min/max* functions return only one value. This leads to a value dominance of one of the two input values while the other one is completely ignored.

The algebraic product $a \cdot b$ for \wedge and the algebraic sum $a + b - a \cdot b$ for \vee , which overcomes the dominance problem of *min/max*, has been also proposed in fuzzy logic [18]. However, a large number of logical laws and semantically equivalences are known from Boolean logic. A user who is intuitively familiar with this equivalences would expect that the same rules are still valid in fuzzy and quantum logic. Unfortunately, in fuzzy logic the algebraic product is not idempotent² and thus no distributivity⁴ holds.

This can be demonstrated by the following example. Let us assume we have a table with an image attribute. Be t_1 a tuple: $t_1[im] = (ref1)$. Further let us assume, a request is composed of a logical combination of a condition with itself, e. g., $c_2 \equiv QBM_{similar}(im) \wedge QBM_{similar}(im)$. This request should produce the same result as the evaluation of a single condition⁷. To evaluate a similarity condition $QBM_{similar}$, we need score values for each fuzzy set and predicate, e.g. $\mu_{[QBM_{similar}(im)]}(t_1)$ or $\varphi(t_1, QBM_{similar}(im))$. We set the score value 1.0, if the queried property is rated by the best possible mark 1. In our example the evaluation of $QBM_{similar}(im)$ for t_1 is simulated and results in the score 0.7.

$$eval_{F/prod}(t, c_2) = \mu_{[QBM_{similar}(a_1)]}(t) * \mu_{[QBM_{similar}(a_1)]}(t) \tag{11}$$

$$= (\mu_{[QBM_{similar}(a_1)]}(t))^2 \tag{12}$$

$$eval_Q(t, c_2) = eval_Q(t, \mathbf{norm}(QBM_{similar}(a_i) \wedge QBM_{similar}(a_i))) \tag{13}$$

$$= \varphi(t, QBM_{similar}(a_i)) \tag{14}$$

Thus, referring to the user expectation we achieve an incorrect result $eval_{F/prod}(t_1, c_2) = 0.49 \neq 0.7 = eval_{F/prod}(t_1, QBM_{similar}(im))$ in fuzzy logic. In contrast to common fuzzy logic, the quantum logic is able to compute the correct result $eval_Q(t_1, c_2) = 0.7$, because of its normalization algorithm (Eq. [13]) recognizes that the underlying condition space (see [13]) for ' $QBM_{similar}(im)$ ' is intersected by itself. Therefore, the operation ' $QBM_{similar}(im) \wedge QBM_{similar}(im)$ ' can be simplified to ' $QBM_{similar}(im)$ ' before any evaluation rule is applied (Eq. [14]).

Contrarily, the quantum logic approach is able to differentiate semantical cases by applying Boolean transformation rules on vector spaces during the applied

normalization. This is impossible in fuzzy logic because required semantics are hidden behind the membership values of the given fuzzy sets [22].

5.4 Example

The following section demonstrate the use of the MPQF algebra and its transformation process on a multimeida request targeting on MPEG-7 metadata and their repository (for instance [6]). Let us assume the imaginary repository stores images and related information like *Title*, *TextAnnotation* and *FileSize* in MPEG-7. See Table [1] for an example data set.

Table 1. Test data set

sv	.../Title	.../TextAnnotation	.../FileSize	/MediaUri
1	Title 01	City:Berlin	1245	MediaUri123
1	Title 12	City:Cottbus	3245	MediaUri23
1	Title 04	City:Passau	1445	MediaUri1323
1	Title 02	City:Paris	945	MediaUri122

Related to the test data, the given request could be considered: *Give me all images and their title that are similar to the given example image and that have been taken in Berlin whereby the fulfilling of the retrieval condition is more important than an association to Berlin. Furthermore, the file size of a result image must not exceed 2048K.*

Based on the underlying metadata model (namely MPEG-7) and the use of MPQF, the request can be formulated as demonstrated in Code [1]. By following the mapping guidelines we can build a MPQF algebra expression: $E \equiv \pi_{.../Title,.../MediaUri}(\sigma_F(R))$. This expression contains a combination of a projection and a selection operation based on a single relation which is extracted from the queried XML document. Furthermore, the selection condition is given by

$$F = ((QBM_{similar}(\dots) \vee \neg 0.8) \wedge (QBFT(\dots) \vee \neg 0.2)) \wedge \dots /FileSize < 2048.$$

To integrate the weighting constraints we have introduced two weighting variables $\theta_{QBM} = 0.8$ and $\theta_{QBFT} = 0.2$ and applied Substitution rule [1] of Section [4.3]. As underlying evaluation model we employ the quantum logic based approach. Then, for instance, considering the first tuple of Table [1] we assume following scores: $\text{eval}(t_1, QB M(\dots))=0.6$ and $\text{eval}(t_1, QBFT(\dots))=0.9$. Moreover, the score for $\text{eval}(t_1, /FileSize < 2048)=1$ is evaluated on a Boolean true/false basis where 0 denotes false and 1 denotes a true. Finally, the result score is gained by computing the following arithmetic expression:

$$\text{eval}(t_1, F) = (0.6 + 0.2 - 0.12) * (0.9 + 0.8 - 0.72) * 1 = 0.68 * 0.98 = 0.6664.$$

5.5 Advantages and Drawbacks

The MPEG Query Format is a very young standard and aims on supporting the access to heterogeneous multimedia databases in an distributed environment. By this, one of the main advantages (see also [7]) of the novel query language is the combination of the expressive style of information as well as XML data retrieval systems. This applies that a query request may feature exact matches as well as fuzzy operations at the same time. Another highlight is the multitude of typical multimedia specific operations including spatial, temporal or example based searches which is absolutely novel in this area. This is further enriched by providing means for assigning weighting and threshold parameters as well as the selection of individual scoring functions for evaluation. Besides, the query language is data model agnostic and supports any XML based metadata format. Finally, the management component especially highlights its use in an distributed scenario for accessing multiple multimedia databases.

Although, the query language has many merits, there are some drawbacks. First of all, the query language itself does not contain any data manipulation functionality such as delete, insert or update operations. However, this

Code 1. Example Request in MPQF

```

<MpegQuery><Query><Input>
  <OutputDescription>
    <ReqField
      typeName="CreationInformationType"/>Creation/Title
    </ReqField>
    <ReqField
      typeName="MediaLocatorType"/>MediaUri
    </ReqField>
  </OutputDescription>
  <QueryCondition>
    <Condition xsi:type="AND">
      <Condition xsi:type="AND">
        <Condition xsi:type="QueryByMedia"
          preferenceValue="0.8" matchType="similar">
          <MediaResource resourceID="ID1">
            <MediaResource>
              <MediaUri>URI_to_Example_Image</MediaUri>
            </MediaResource>
          </MediaResource>
        </Condition>
        <Condition xsi:type="QueryByFreeText"
          preferenceValue="0.2">
          <FreeText>City:Berlin</FreeText>
        </Condition>
      </Condition>
    </Condition>
    <Condition xsi:type="SmallerThan">
      <ArithmeticField typeName="MediaFormatType">
        /FileSize
      </ArithmeticField>
      <LongValue>2048</LongValue>
    </Condition>
  </Condition>
</Input></Query></MpegQuery>

```

circumstance can also be found at other well known query languages like XQuery (XML data) or SPARQL (semantic data). But more problematic is the absence of necessary parameters in the specification of some query types. For instance, the *QueryByMedia* query type allows only to add the example media but does not support to assign the target element in the data model to be evaluated against. This means that a complex query request where two different target (user selected) images should be addressed is not possible. Finally, the available filter mechanisms offer a rich set of selection possibilities but is limited by the use and combination of multiple sets. That is, the provided join capability only allows the combination of a maximum of two sets in one single query request. Furthermore, features such as subqueries are not supported by the MPEG Query Format at all.

6 Conclusion and Future Work

This article presented a formal algebra representation of the novel MPEG Query Format. This includes the description of the data model (based on an extension of XDM), the specification of the syntax and semantics of the developed algebra and its multimedia operations for accessing multimedia data. Related on our generic evaluation methodology for condition processing different logics for its execution can be chosen. In this context, the article demonstrated the use and the respective advantages and shortcomings of the common fuzzy logic and the novel quantum logic approach. By this, it could be shown that the quantum logic approach provides better characteristics for multimedia retrieval although there are some restrictions for the creation of query instances.

Future work will focus on two main directions. First, endeavors will be made to implement the introduced algebra into a MPEG Query Format aware native database system. Second, methods will be investigated for overcoming the identified problems, such as missing extensible join functionality, missing parameters in some query types, etc. This investigation aims on a refinement and improvement of the first version of the standard.

Acknowledgement. This work has been partially supported by the THESEUS Program and the SQ-System project which are funded by the German Federal Ministry of Economics and Technology and the DFG funding association.

References

1. Adali, S., Bonatti, P.A., Sapino, M.L., Subrahmanian, V.S.: A Multi-Similarity Algebra. *ACM Sigmod Record* 27(2), 402–413 (1989)
2. Atnafu, S., Brunie, L., Kosch, H.: Similarity-Based Algebra for Multimedia Database Systems. In: *Proceedings of the 12th ACM Australasian Database Conference*, Gold Coast, Queensland, Australia, pp. 115–122 (2001)
3. Belohlavek, R., Optichal, S., Vychodil, V.: Relational algebra for ranked tables with similarities: Properties and implementation. In: Berthold, M., Shawe-Taylor, J., Lavrač, N. (eds.) *IDA 2007*. LNCS, vol. 4723, pp. 140–151. Springer, Heidelberg (2007)

4. Brantner, M., Helmer, S., Kanne, C.-C., Moerkotte, G.: Full-fledged Algebraic XPath Processing in Natix. In: Proceedings of the 21st International Conference on Data Engineering, pp. 705–716 (2005)
5. Ciaccia, P., Montesi, D., Penzo, W., Trombetta, A.: Imprecision and user preferences in multimedia queries: A generic algebraic approach. In: Schewe, K.-D., Thalheim, B. (eds.) FoIKS 2000. LNCS, vol. 1762, pp. 50–71. Springer, Heidelberg (2000)
6. Döller, M., Kosch, H.: The MPEG-7 Multimedia Database System (MPEG-7 MMDB). *Journal of Systems and Software* 81(9), 1559–1580 (2008)
7. Döller, M., Tous, R., Gruhne, M., Yoon, K., Sano, M., Burnett, I.S.: The MPEG Query Format: On the way to unify the access to Multimedia Retrieval Systems. *IEEE Multimedia* 15(4), 82–95 (2008)
8. Fagin, R., Wimmers, E.L.: A Formula for Incorporating Weights into Scoring Rules. *Theoretical Computer Science* 239(2), 309–338
9. Jagadish, H.V., Lakshmanan, L.V.S., Srivastava, D., Thompson, K.: TAX: A Tree Algebra for XML. In: Proceedings of 8th International Workshop on Databases and Programming Languages, Rome, Italy, pp. 149–164 (2001)
10. Klose, A., Nürnberger, A.: On the Properties of Prototype-Based Fuzzy Classifiers. *IEEE Transaction on Systems, Man and Cybernetics* 37(4), 817–835 (2007)
11. Kosch, H., Atnafu, S.: A Multimedia Join by the Method of Nearest Neighbor Search. *Information Processing Letters (IPL)* 82(5), 269–276 (2002)
12. Lee, J.H.: Properties of Extended Boolean Models in Information Retrieval. In: SIGIR (ed.) Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, pp. 182–190. Springer-Verlag New York, Inc., Secaucus (1994)
13. Lehrack, S., Schmitt, I.: The theoretical model behind CQQL. Technical report, BTU Cottbus (2009)
14. Lin, H., Huang, B.: SQL/SDA: A Query Language for Supporting Spatial Data Analysis and Its Web-Based Implementation. *IEEE Transactions on Knowledge And Data Engineering* 13(4), 671–682 (2001)
15. Martinez, J.M., Koenen, R., Pereira, F.: MPEG-7. *IEEE Multimedia* 9(2), 78–87 (2002)
16. Moro, M.M., Edelweiss, N., Zaupa, A.P., dos Santos, C.S.: TVQL - temporal versioned query language. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, pp. 618–627. Springer, Heidelberg (2002)
17. Paparizos, S., Wu, Y., Lakshmanan, L.V.S., Jagadish, H.V.: Tree Logical Classes for Efficient Evaluation of XQuery. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp. 71–82 (2004)
18. Rudolf Kruse, J.E.G., Klawonn, F.: Foundations of Fuzzy Systems, 278 pages (1994); ISBN-10: 047194242X
19. Schering, A.-C., Balouch, A.S., Heuer, A.: BSA-Algebra für XQuery, Operation - Optimierungsregeln und Anwendungen. In: Proceedings of the 18th GI-Workshop on the Foundations of Databases (Grundlagen von Datenbanken), pp. 135–139 (2006)
20. Schmitt, I.: Weighting in CQQL. Technical report, Brandenburg University of Technology at Cottbus, Institute of Computer Science (2007)
21. Schmitt, I.: QQL: A DB&IR Query Language. *The VLDB Journal* 17(1), 39–56 (2008)

22. Schmitt, I., Nuernberger, A., Lehrack, S.: On the Relation between Fuzzy and Quantum Logic. In: Views on Fuzzy Sets and Systems from Different Perspectives. Philosophy and Logic, Criticisms and Applications, pp. 421–432 (2009)
23. Schmitt, I., Schulz, N.: Similarity relational calculus and its reduction to a similarity algebra. In: Seipel, D., Turull-Torres, J.M.a. (eds.) FoIKS 2004. LNCS, vol. 2942, pp. 252–272. Springer, Heidelberg (2004)
24. Xue, L., Li, C., Wu, Y., Xiong, Z.: VeXQuery: An xQuery extension for MPEG-7 vector-based feature query. In: Damiani, E., Yetongnon, K., Chbeir, R., Dipanda, A. (eds.) SITIS 2006. LNCS, vol. 4879, pp. 34–43. Springer, Heidelberg (2009)
25. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Author Index

- Batko, Michal 159
Baumann, Stephan 1
Brut, Mihaela 31
Budikova, Petra 159
Bürger, Tobias 61
- Catarci, Tiziana 16
Chan, Ching Hau 103
- de Vries, Arjen P. 89
Dohnal, Vlastislav 174
Döller, Mario 61, 204
- Erra, Ugo 189
- Homola, Tomas 174
- Ioannidis, Yannis 16
Ionescu, Bogdan E. 74
- Jones, Gareth J.F. 103
- Katifori, Akrivi 16
Knees, Peter 118
Kosch, Harald 61, 204
Koutrika, Georgia 16
- Lambert, Patrick 74
Lehrack, Sebastian 204
- Manola, Natalia 16
- Nika, Ana 16
Nürnberger, Andreas 16, 144
- Rasche, Christoph 74
Rauber, Andreas 132
Rode, Henning 89
- Schindler, Alexander 132
Schirru, Rafael 1
Schmitt, Ingo 46, 204
Sedes, Florence 31
Senatore, Sabrina 189
Seyerlehner, Klaus 118
Stegmaier, Florian 61
Stober, Sebastian 144
Streit, Bernhard 1
- Thaller, Manfred 16
Tsikrika, Theodora 89
- Vertan, Constantin 74
- Widmer, Gerhard 118
- Zellhöfer, David 46
Zezula, Pavel 159, 174