

**Marcin Detyniecki Joemon M. Jose
Andreas Nürnberger C.J. van Rijsbergen (Eds.)**

LNCS 3877

Adaptive Multimedia Retrieval: User, Context, and Feedback

**Third International Workshop, AMR 2005
Glasgow, UK, July 2005
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

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

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Marcin Detyniecki
Joemon M. Jose
Andreas Nürnberger
C.J. van Rijsbergen (Eds.)

Adaptive Multimedia Retrieval: User, Context, and Feedback

Third International Workshop, AMR 2005
Glasgow, UK, July 28-29, 2005
Revised Selected Papers

Volume Editors

Marcin Detyniecki
Laboratoire d'Informatique de Paris 6, LIP6
8 rue du Capitaine Scott, 75015 Paris, France
E-mail: Marcin.Detyniecki@lip6.fr

Joemon M. Jose
C.J. van Rijsbergen
University of Glasgow
Department of Computing Science
University Avenue, Glasgow G12 8QQ, UK
E-mail: {jj,keith}@dcs.gla.ac.uk

Andreas Nürnberger
Otto-von-Guericke Universität Magdeburg
Fakultät für Informatik
Universitätsplatz 2, 39106 Magdeburg, Germany
E-mail: nuernb@iws.cs.uni-magdeburg.de

Library of Congress Control Number: 2006920781

CR Subject Classification (1998): H.3, H.5.1, H.5.5, I.4, I.2

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

ISSN 0302-9743
ISBN-10 3-540-32174-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-32174-3 Springer Berlin Heidelberg New York

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.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11670834 06/3142 5 4 3 2 1 0

Preface

This book is an extended collection of revised contributions that were initially submitted to the International Workshop on Adaptive Multimedia Retrieval (AMR 2005). This workshop was organized during July 28-29, 2005, at the University of Glasgow, UK, as part of an information retrieval research festival and in co-location with the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005). AMR 2005 was the third and so far the biggest event of the series of workshops that started in 2003 with a workshop during the 26th German Conference on Artificial Intelligence (KI 2003) and continued in 2004 as part of the 16th European Conference on Artificial Intelligence (ECAI 2004).

The workshop focussed especially on intelligent methods to analyze and structure multimedia collections, with particular attention on methods that are able to support the user in the search process, e.g., by providing additional user- and context-adapted information about the search results as well as the data collection itself and especially by adapting the retrieval tool to the user's needs and interests. The invited contributions presented in the first section of this book—"Putting the User in the Loop: Visual Resource Discovery" from Stefan Ruger, "Using Relevance Feedback to Bridge the Semantic Gap" from Ebroul Izquierdo and Divna Djordjevic, and "Leveraging Context for Adaptive Multimedia Retrieval: A Matter of Control" from Gary Marchionini—illustrate these core topics: user, context and feedback. These aspects are discussed from different points of view in the 18 contributions that are classified into six main chapters, following rather closely the workshop's sessions: ranking, systems, spatio-temporal relations, using feedback, using context and meta-data. We think that this book provides a good and conclusive overview of the current research in this area.

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 and Alfred Hofmann from Springer for his support in publishing this book.

October 2005

Marcin Detyniecki, Paris, France
Joemon M. Jose, Glasgow, UK
Andreas Nurnberger, Magdeburg, Germany
Keith van Rijsbergen, Glasgow, UK

Sponsoring Institutions

MMKM – Multimedia Knowledge Management Network, UK

University of Glasgow, UK

BCS – The British Computer Society, UK

British HCI Group, UK

UK Sharp Laboratories of Europe Ltd.

Organization

General Chair

Keith van Rijsbergen University of Glasgow, UK

Program Chairs

Andreas Nürnberger University of Magdeburg, Germany
Marcin Detyniecki CNRS, Lab. d'Informatique de Paris 6, France
Joemon M. Jose University of Glasgow, UK

Local Chair

Iadh Ounis University of Glasgow, UK

Publicity Chair

Jana Urban University of Glasgow, UK

Program Committee

Jenny Benois-Pineau University of Bordeaux, LABRI, France
Pia Borlund Royal School of Libr. and Inform. Science,
Denmark
Arjen De Vries CWI, Amsterdam, The Netherlands
Norbert Fuhr University of Duisburg-Essen, Germany
Bogdan Gabrys Bournemouth University, UK
Ana M. García Serrano Universidad Politécnica de Madrid, Spain
Sebastian Goeser IBM Germany Development Ltd., Germany
Philippe Joly Université Paul Sabatier, Toulouse, France
Gareth Jones Dublin City University, Ireland
Pietro Pala Università di Firenze, Italy
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 Sophia Antipolis Cédex, France
Gheorghe Muresan Rutgers University, USA
Stefan Rieger Imperial College London, UK

VIII Organization

Nicu Sebe
Ingo Schmitt
Alan F. Smeaton

Leiden University, The Netherlands
University of Magdeburg, Germany
Dublin City University, Ireland

Table of Contents

Invited Contributions

Putting the User in the Loop: Visual Resource Discovery <i>Stefan Ruger</i>	1
Using Relevance Feedback to Bridge the Semantic Gap <i>Ebroul Izquierdo, Divna Djordjevic</i>	19
Leveraging Context for Adaptive Multimedia Retrieval: A Matter of Control <i>Gary Marchionini</i>	35

Ranking

Rank-Ordering Documents According to Their Relevance in Information Retrieval Using Refinements of Ordered-Weighted Aggregations <i>Mohand Boughanem, Yannick Loiseau, Henri Prade</i>	44
Ranking Invariance Based on Similarity Measures in Document Retrieval <i>Jean-Francois Omhover, Maria Rifqi, Marcin Detyniecki</i>	55

Systems

Developing AMIE: An Adaptive Multimedia Integrated Environment <i>Osama El Demerdash, Sabine Bergler, Leila Kosseim, P. Karen Langshaw</i>	65
Exploring the Structure of Media Stream Interactions for Multimedia Browsing <i>Saturnino Luz, Matt-Mouley Bouamrane</i>	79
CARSA - An Architecture for the Development of Context Adaptive Retrieval Systems <i>Korinna Bade, Ernesto W. De Luca, Andreas Nurnberger, Sebastian Stober</i>	91
Integrating Media Management Towards Ambient Intelligence <i>Willem Fontijn, Jan Nesvadba, Alexander Sinitsyn</i>	102

CANDELA - Storage, Analysis and Retrieval of Video Content in Distributed Systems
E. Jaspers, R. Wijnhoven, R. Albers, J. Nesvadba, J. Lukkien, A. Sinitsyn, X. Desurmont, P. Pietarila, J. Palo, R. Truyen 112

Spatio-temporal Relations

Interactive Retrieval of Video Sequences from Local Feature Dynamics
Nicolas Moënné-Loccoz, Eric Bruno, Stéphane Marchand-Maillet 128

Temporal Relation Analysis in Audiovisual Documents for Complementary Descriptive Information
Zein Al Abidin Ibrahim, Isabelle Ferrane, Philippe Joly 141

Using Feedback

Using Segmented Objects in Ostensive Video Shot Retrieval
Sorin Sav, Hyowon Lee, Alan F. Smeaton, Noel E. O'Connor 155

Learning User Queries in Multimodal Dissimilarity Spaces
Eric Bruno, Nicolas Moenne-Loccoz, Stéphane Marchand-Maillet 168

Surface Features in Video Retrieval
Thijs Westerveld, Arjen P. de Vries, Georgina Ramírez 180

Toward Consistent Evaluation of Relevance Feedback Approaches in Multimedia Retrieval
Xiangyu Jin, James French, Jonathan Michel 191

Using Context

An Explorative Study of Interface Support for Image Searching
Jana Urban, Joemon M. Jose 207

Context-Based Image Similarity Queries
Ilaria Bartolini 222

Meta Data

Information Retrieval of Sequential Data in Heterogeneous XML Databases
Eugen Popovici, Pierre-François Marteau, Gildas Ménier 236

A Visual Annotation Framework Using Common-Sensical and Linguistic Relationships for Semantic Media Retrieval <i>Bageshree Shevade, Hari Sundaram</i>	251
Improving Access to Multimedia Using Multi-source Hierarchical Meta-data <i>Trevor P. Martin, Yun Shen</i>	266
Author Index	279

Putting the User in the Loop: Visual Resource Discovery

Stefan Ruger

Department of Computing, South Kensington Campus,
Imperial College London, London SW7 2AZ, UK
s.rueger@imperial.ac.uk

Abstract. Visual resource discovery modes are discussed with a view to apply them in a wide variety of digital multimedia collections. The paradigms include summarising complex multimedia objects such as TV news, information visualisation techniques for document clusters, visual search by example, relevance feedback and methods to create browsable structures within the collection. These exploration modes share three common features: they are automatically generated, depend on visual senses and interact with the user of the multimedia collections.

1 Introduction

Giving users access to collections is one of the defining tasks of a library. For thousands of years the traditional methods of resource discovery have been searching, browsing and asking: Librarians create reference cards with meta-data that are put into catalogues (nowadays databases); they also place the objects in physical locations that follow certain classification schemes and they answer questions at the reference desk.

The advent of digital documents has radically changed the organisation principles: Now it is possible to *automatically* index and search document collections as big as the world-wide web *à la* Google and browse collections utilising author-inserted links. It is almost as if automated processing has turned the traditional library access upside down: instead of searching meta-data catalogues in order to retrieve the document, web search engines search the full content of documents and retrieve their meta-data, ie, the location where documents can be found. Undoubtedly, this automated approach has made all the difference to the way the vast world-wide web can be utilised.

However, indexing sheer mass is no guarantee of success either: While most of today's inter-library loan systems allow access to virtually any publication in the world (at least to around 40m entries in OCLC's Worldcat database and a further 3m from Bowker's Books In Print), students and researchers alike seem to be reluctant to actually make use of this facility. On the other hand, the much smaller catalogue offered by Amazon appears to be very popular — presumably owing to added services such as subject categories; fault tolerant search tools; personalised services telling the customer what's new in a subject area or what

other people with a similar profile bought; pictures of book covers; media and customer reviews; access to the table of contents, to selections of the text and to the full-text index of popular books; and the perception of fast delivery.

This paper argues that automated added services such as visual queries, browsing and summaries can prove useful for resource discovery in multimedia digital libraries. Multimedia collections pose their very own challenges in this context; images and videos don't usually come with dedicated reference cards or meta-data, and when they do, as in museum collections, their creation will have been expensive and time-consuming. The next section explores methods of automatically indexing, labelling and annotating image and video content. It briefly discusses the challenges of the semantic gap, polysemy, fusion and responsiveness inherent with these. Sections 3 and 4 are about summarising techniques for videos and about visualisation of search results, while Section 5 discusses content-based visual search modes such as query by example and relevance feedback. Section 6 promotes browsing as resource discovery mode and looks at underlying techniques to automatically structure the document collection.

2 Challenges of Automated Visual Indexing

Videos can be annotated, ie, get indexable text strings assigned, using a variety of sources: closed-captions, teletext, subtitles, automated speech recognition on the audio and optical character recognition for text embedded in the frames of a video. The resulting text strings are then used as the basis for full-text indexing, which is the way most video retrieval systems operate, including Google's latest TV search engine <http://video.google.com>.

Automatically annotating images with text strings is less straightforward. Methods attempting this task include dedicated machine vision models for particular words (such as 'people' or 'aeroplane'); machine translation methods that link image regions (blobs) and words in the same way as corresponding words in two text documents written in different languages but otherwise of same contents [30]; co-occurrence models of low-level image features of tiled image regions and words [52]; cross-lingual information retrieval models [46, 48]; inference networks that connect image segments with words [51]; probabilistic modelling with Latent Dirichlet Allocation [9], Bernoulli distributions [31] or non-parametric density estimation [81]; Support Vector Machine classification and relevance feedback [45]; and simple scene-level statistics [72]. All these methods have in common that a controlled vocabulary set of limited size (in the order of 500 more or less general terms) is used to annotate images based on a large training set.

The commonest way of indexing the visual content of images is by extracting low-level features, which represent colour usage, texture composition, shape and structure, localisation or motion. These representations are often real-valued vectors containing summary statistics, eg, in the form of histograms; their respective distances act as indicators whether or not two images are similar with respect to this particular feature. Design and usage of these features can be

critical, and there is a wide variety of them, eg, as published by participants in the TRECVID conference [74, 34]. Once created, those features will allow the comparison and ranking of images in the database with respect to images submitted as a query (*query-by-example*).

There are a number of open issues with this approach: On a perceptual level, those low-level features do not necessarily correlate with any high-level meaning the images might have, such as victory or triumph. Even if they did, images usually convey a multitude of meanings so that the query-by-example approach is bound to under-specify the real information need. The former problem is known as *semantic gap* and the latter as *polysemy*. Designing a human-computer interaction that utilises the user's feedback has been one of the main approaches to tackle these perceptual issues. Amongst other methods there are those that seek to reformulate the query [44, 49, 58] or those that weight the various features differently depending on the user's feedback. Weight adaptation methods include non-parametric density estimation [50]; cluster analysis of the images [79]; transposed files for feature selection [71]; Bayesian network learning [21, 50]; statistical analysis of the feature distributions of relevant images [60]; variance analysis [60]; and analytic global optimisation [36, 40]. Some approaches give the presentation and placement of images on screen much consideration to indicate similarity of images amongst themselves [64, 59] or with respect to a visual query [36, 40].

On a practical level, the multitude of features assigned to images poses a *fusion problem*: how to combine possibly conflicting evidence of two images' similarity? There are many approaches to carry out fusion, some based on labelled training data and some based on user feedback for the current query [3, 7, 67, 80].

There is a *responsiveness problem*, too, in that the naïve comparison of query feature vectors to the database feature vectors requires a linear scan through the database. Although the scan is eminently scalable, the practicalities of doing this operation can mean an undesirable response time in the order of seconds rather than the 100 milli-seconds that can be achieved by text search engines. The problem is that high-dimensional tree structures tend to collapse to linear scans above a certain dimensionality [77]. As a consequence, some approaches for fast nearest-neighbour search use compression techniques to speed up the disk access of linear scan as in [77] using VA-files; or they approximate the search [54, 8]; decompose the features componentwise [28, 1, 14] saving access to unnecessary components; or deploy a combination of these [53, 43].

3 Video Summaries

Even if the automated methods of the preceding section enabled a retrieval process with high precision (proportion of the retrieved items that are relevant) and high recall (proportion of the relevant items that are retrieved) it would still be vital to present the retrieval results in a way so that the users can quickly decide whether or not those items are relevant to them.

Images are most naturally displayed as thumbnails, and their relevance can quickly be judged by users. Presenting and summarising videos is a bit more

involved. The main metaphor used for this is that of a *storyboard* that would contain *keyframes* with some text about the video. Several systems exist that summarise news stories in this way, most notably Informedia [18] and Físchlár [69]. The Informedia system devotes much effort to added services such as face recognition and speaker voice identification allowing retrieval of the appearance of known people. Informedia also provides alternative modes of presentation, eg, through film skims or by assembling ‘collages’ of images, text and other information (eg, maps) sourced via references from the text [17]. Físchlár’s added value lies in the ability to personalise the content (with the user expressing like or dislike of stories) and in assembling lists of related stories and recommendations.

Our very own TV news search engine ANSES [57,56] records the main BBC evening news along with the sub-titles, indexes them, breaks the video stream

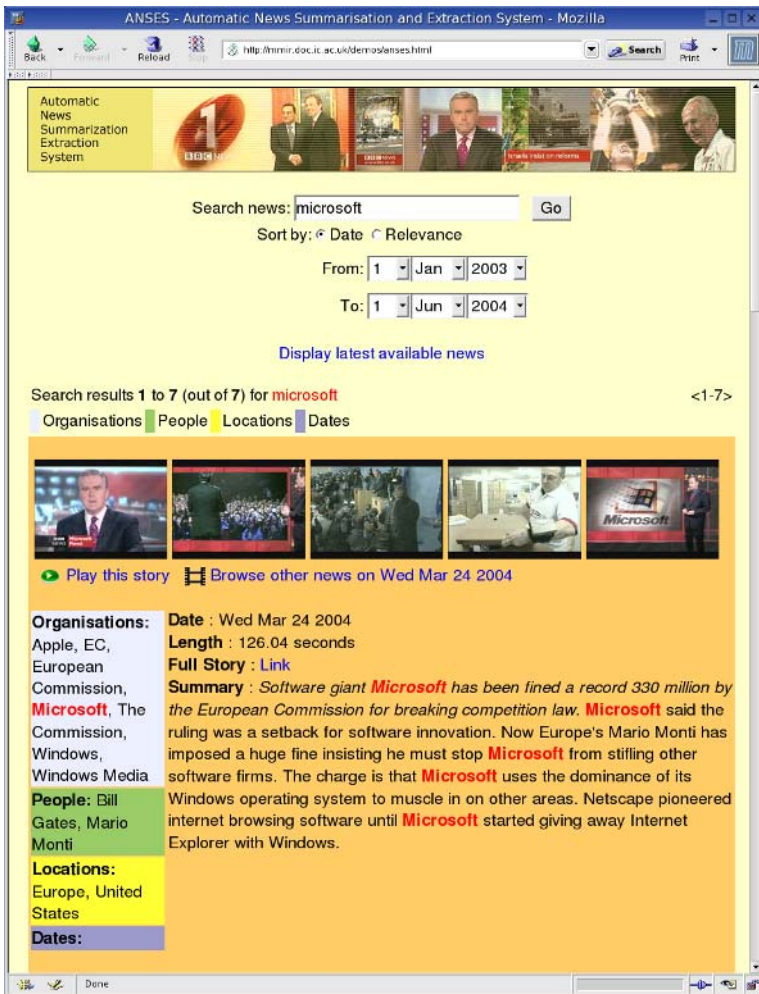


Fig. 1. News search engine interface

into shots (defined as those video sequences that are generated during a continuous operation of the camera), extracts one key-frame per shot, automatically glues shots together to form news stories based on an overlap in vocabulary in the sub-titles of adjacent shots (using lexical chains), and assembles a story-board for each story that can be retrieved using text searches or browsed. Fig 1 shows the interface of ANSES. The natural language toolset GATE [26, 27] is used for automated discovery of organisations, people, places and dates to provide an instant indication of what the news story is about. ANSES also displays a short automated textual extraction summary, again using lexical chains to identify the most salient sentences. These summaries are never as informative as hand-made ones, but users of the system have found them crucial for judging whether or not they are interested in a particular returned search result.

Dissecting the video stream into shots and associating one keyframe along with text from subtitles to each shot has another advantage: A video collection can essentially be treated as an image collection, where each, possibly annotated, image acts as entry point into the video.

4 New Paradigms in Information Visualization

The last decade has witnessed an explosion in interest in the field of information visualization, e.g. [47, 15, 70, 41, 2, 12, 55, 4, 68, 82, 66, 10]. We added three new techniques to the pool of existing visualization paradigms, based on our design studies [5, 13]. These techniques all revolve around a representation of documents in the form of bag-of-words vectors, which can be clustered to form groups; we use a variant of the buckshot clustering algorithm for this. Another common element of our visualisations is the notion of *keywords* that are specific to the returned set of documents. The keywords are computed using a simple statistic; for details see [13, 39]. The new methods are:

Sammon Cluster View. This paradigm uses a Sammon map to generate a two dimensional screen location from a many-dimensional vector representing a cluster centroid. This map is computed using an iterative gradient search [63] while attempting to preserve the pairwise distances between the cluster centres. Clusters are thus arranged that their mutual distances are indicative of their relationship. The idea is to create a visual landscape for navigation. Fig 2 shows an example of such an interface. The display has three panels, a scrolling table panel to the left, a graphic panel in the middle and a scrolling text panel to the right that contains the traditional list of returned documents as hotlinks and snippets. In the graphic panel each cluster is represented by a circle and is labelled with its two most frequent keywords. The radius of the circle informs about the cluster size. The distance between any two circles in the graphic panel is an indication of the similarity of their respective clusters: the nearer the clusters, the more likely the documents contained within will be similar. When the mouse passes over the cluster circle a ‘tool tip’ box in the form of a pop-up menu appears that allows the user to select clusters and *drill down*, ie, re-cluster and re-display only the documents in the selected clusters. The back button undoes this process and

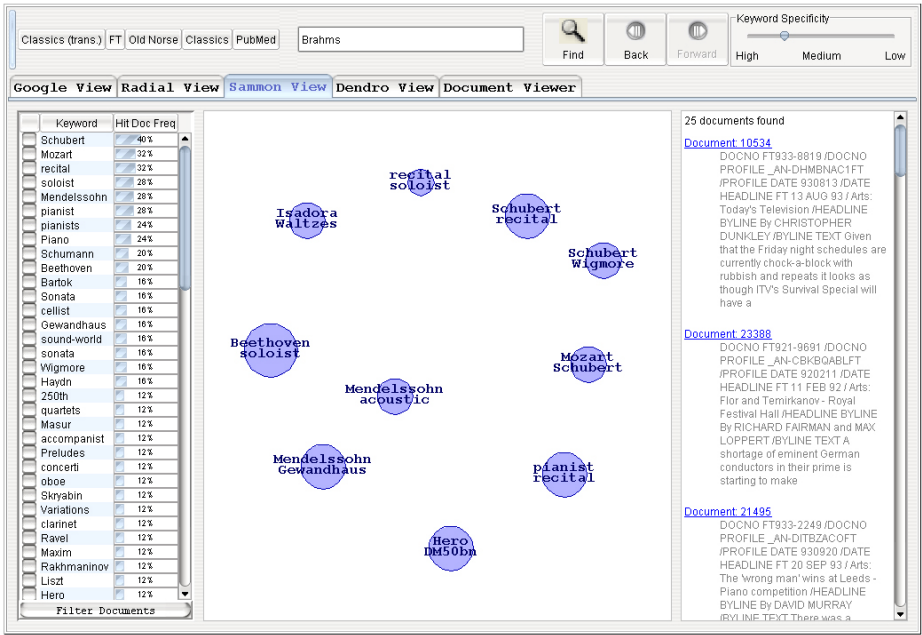


Fig. 2. Sammon map for cluster-guided search

climbs up the hierarchy (*drill up*). The table of keywords includes box fields that can be selected. At the bottom of the table is a filter button that makes the scrolling text window display only the hot-links and snippets from documents that contain the selected keywords.

Dendro Map Visualization. The Dendro Map visualization represents documents as leaf nodes of a binary tree that is output by the buckshot clustering algorithm. With its plane-spanning property and progressive shortening of branches towards the periphery, the Dendro Map mimics the result of a non-Euclidean transformation of the plane as used in hyperbolic maps without suffering from their computational load. Owing to spatial constraints, the visualization depth is confined to five levels of the hierarchy with nodes of the lowest level representing either documents or subclusters. Different colours facilitate visual discrimination between individual documents and clusters. Each lowest level node is labelled with the most frequent keyword of the subcluster or document. This forms a key component of the Dendro Map as it gives the user the cues needed for navigating through the tree. As the user moves the mouse pointer over an internal node, the internal nodes and branches of the associated subcluster change colour from light blue to dark blue while the leaf nodes, ie, document representations, turn bright red. As in the Sammon Map, a tool-tip window provides additional information about the cluster and can be used to display a table with a list of keywords associated with the cluster. The user may drill down on any internal node. The selected node will as a result replace the current root node at the center and

the entire display is re-organized around the new root. The multi-level approach of the Dendro Map allows the user to gain a quick overview over the document collection and to identify promising subsets.

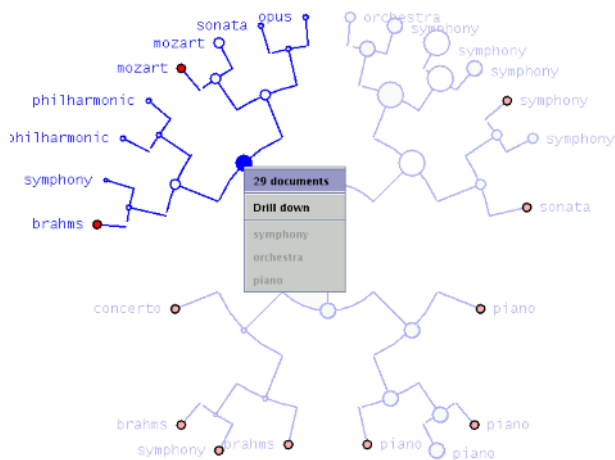


Fig. 3. Dendro Map: a plane-spanning binary tree (query “Beethoven”)

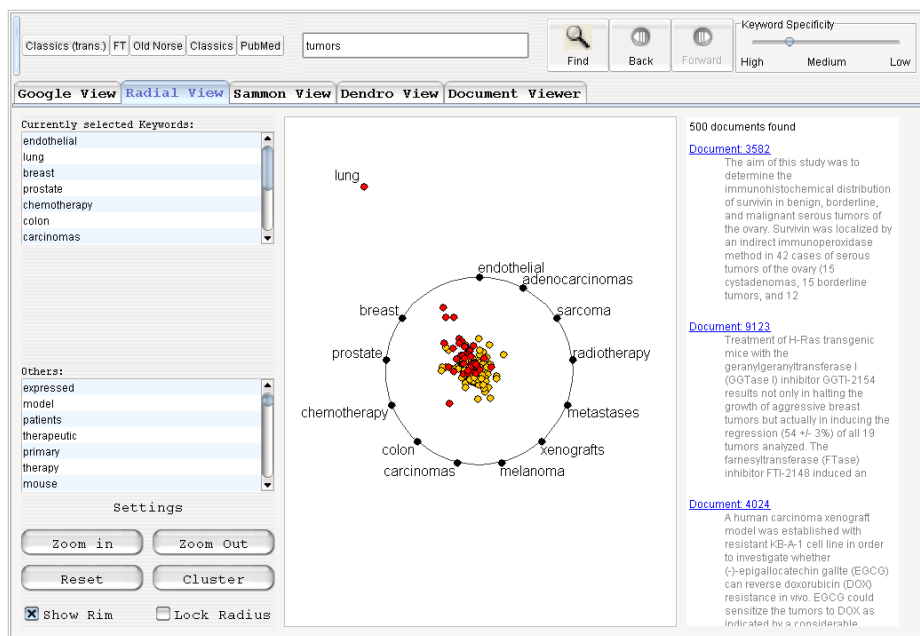


Fig. 4. Radial Visualisation

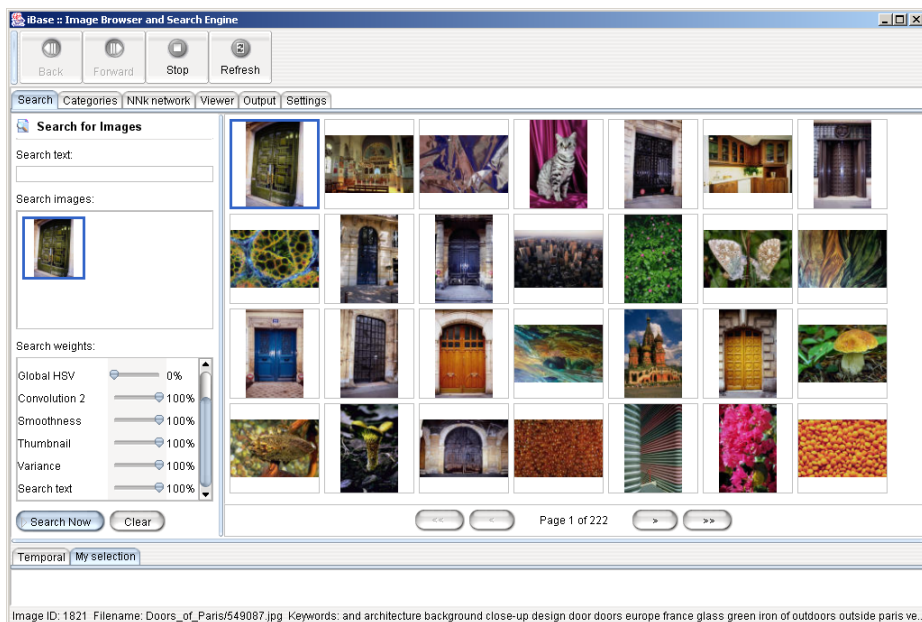
Radial Interactive Visualization. Radial (Figure 4) is similar to VIBE [47], to Radviz [42] and to Lyberworld [41]. It places the keyword nodes round a circle, and the position of the document dots in the middle depend on the force of invisible springs connecting them to keyword nodes: the more relevant a keyword for a particular document, the stronger its spring pulls on the document. Hence, we make direct use of the bag-of-words representation without explicit clustering. Initially, the twelve highest ranking keywords are displayed in a circle. The interface lets the user move the keywords, and the corresponding documents follow this movement. This allows the user to manually cluster the documents based on the keywords they are interested in. As the mouse passes over the documents, a bubble displays a descriptive piece of text. The location of document dots is not unique owing to dimensionality reduction, and there may be many reasons for a document to have a particular position. To mitigate this ambiguity in Radial the user can click on a document dot, and the keywords that affect the location of document are highlighted. A choice of keywords used in the display can be exercised by clicking on two visible lists of words. Zoom buttons allow the degree of projection to be increased or reduced so as to distinguish between documents around the edges of the display or at the centre. The Radial visualization appears to be a good interactive tool to structure the document set according to one’s own preferences by shifting keywords around in the display.

Unified Approach. The integration of the paradigms into one application offers the possibility of browsing the same result set in several different ways simultaneously. The cluster-based visualizations give a broader overall picture of the result, while the Radial visualization allows the user to focus on subsets of keywords. Also, as the clusters are approximations that highlight particular keywords, it may be useful to return to the Radial visualization and examine the effect of these keywords upon the whole document set. The Radial visualization will perhaps be more fruitful if the initial keywords match the user’s area of interest. The Sammon Map will let the user dissect search sets and re-cluster subsets, gradually homing in on target sets. This interface was developed within the joint NSF-EC project CHLT (<http://www.chlt.org>); it was evaluated from a human-computer-interaction point of view with encouraging results [16] and has proven useful in real-world multi-lingual scholarly collections [62].

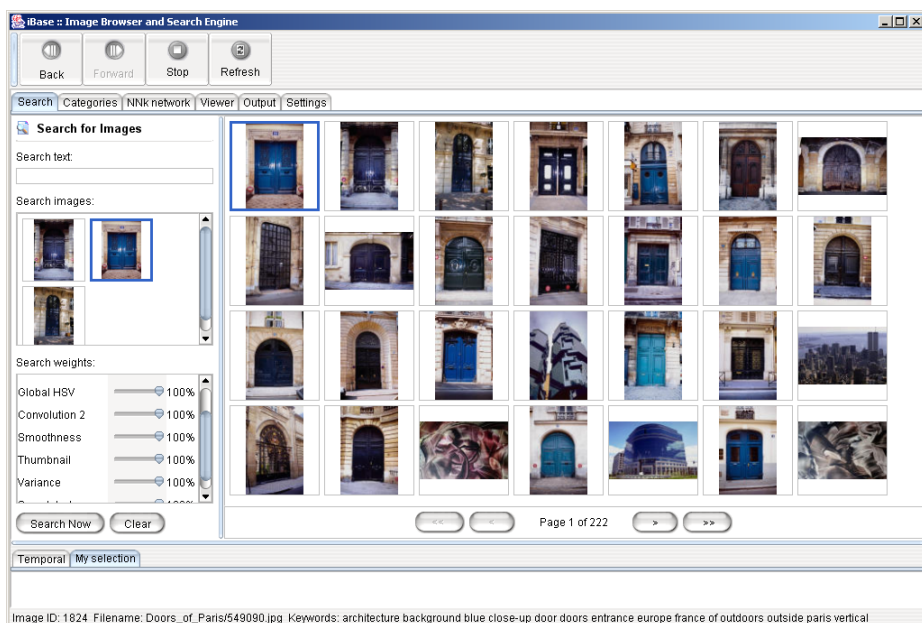
5 Visual Search and Relevance Feedback

The visual query-by-example paradigm discussed in Section 2 gives rise to relatively straightforward interfaces: An image is dragged into a query box, or, eg, specified via a URL, and the best matching images are displayed in a ranked list to be inspected by the user, see Fig 5(a). A natural extension of such an interface is to offer the selection of relevant results as new query elements. This type of relevance feedback, aka *query point moving*, is shown in Fig 5(b).

One other main type of relevance feedback, *weight space movement*, assumes that the relative weight of the multitude of features that one can assign to images



(a) query by example (left panel) with initial results in the right panel



(b) a new query made of three images from (a) results in many more blue-door images

Fig. 5. Visual search for a blue door starting with a green-door example

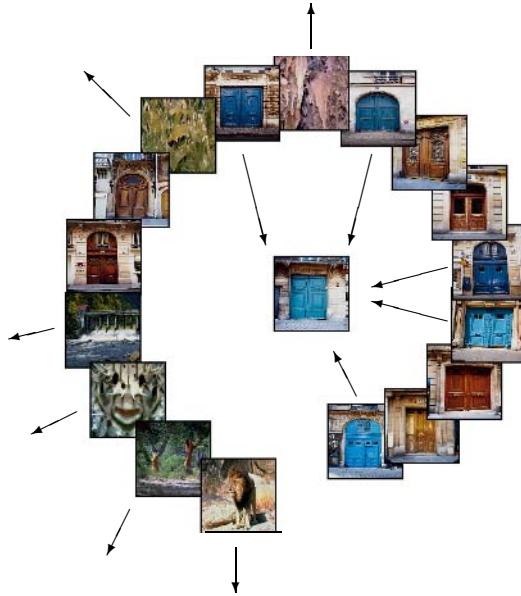


Fig. 6. A relevance feedback model

(eg, structured meta-data fields such as author, creation date and location; low-level visual features such as colour, shape, structure and texture; free-form text) can be learned from user feedback. Of the methods mentioned in Section 2 we chose analytic weight updating as this has a very small execution time. The idea is that users can specify the degree to which a returned image is relevant to their information needs. This is done by having a visual representation: the returned images are listed in a spiral, and the distance of an image to the centre of the screen is a measure of the relevance that the search engine assigns to a specific image. A similar method has subsequently been used in [73]. Users can now move the images around with the mouse or place them in the centre with a left mouse click and far away with a right click. Fig 6 shows this relevance feedback model. We evaluated the effectiveness of negative feedback, positive feedback and query point moving, and found that combining the latter two yields the biggest improvement in terms of mean average precision [36].

6 Browsing: Lateral and Geo-temporal

The idea of representing text documents in a nearest-neighbour network was first presented in [25], albeit, as an internal representation of the relationships between documents and terms, not for browsing. Document networks for interactive browsing were identified by Cox [22, 23]. Attempts to introduce the idea of browsing into content-based image retrieval include Campbell’s work [11]; his ostensive model retains the basic mode of query based retrieval but in addition allows browsing through a dynamically created local tree structure. Jain

and Santini's *El niño* system [65,64] is another attempt to combine query-based search with browsing. The system tries to display configurations of images in feature space such that the mutual distances between images are preserved as best as possible. Feedback is given in the same spirit as in Fig 6 by manually forming clusters of images that appear similar to the user. This in turn results in an altered configuration with potentially new images being displayed.

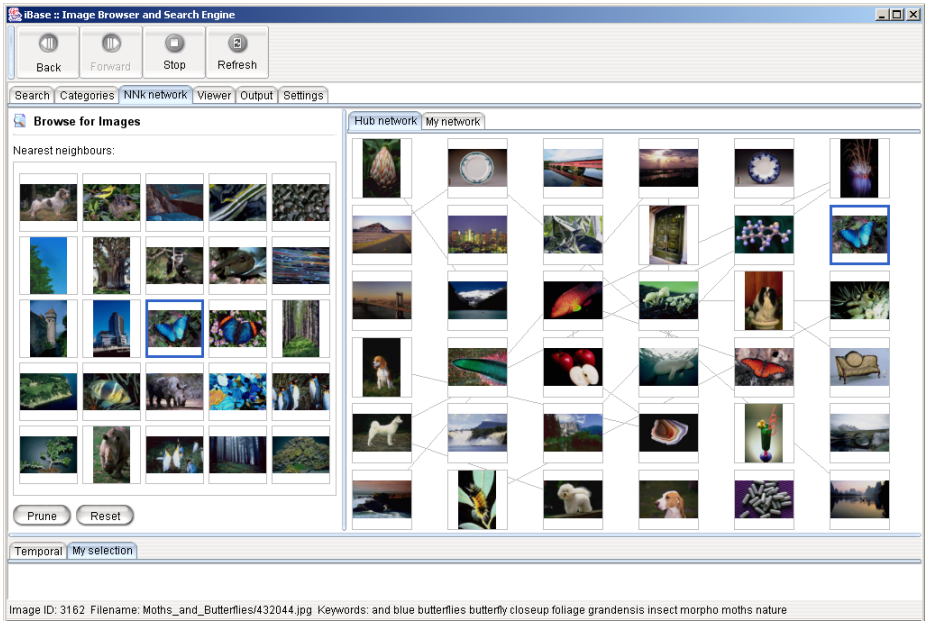
Other Network structures that have increasingly been used for information visualization and browsing are Pathfinder networks [29]. They are constructed by removing redundant edges from a potentially much more complex network. In [32] Pathfinder networks are used to structure the relationships between terms from document abstracts, between document terms and between entire documents. The user interface supports access to the browsing structure through prominently marked high-connectivity nodes.

Our group [37,38,33] determines the nearest neighbour for the image under consideration (which we call the *focal image*) for *every* combination of features. This results in a set of what we call *lateral neighbours*. By calculating the lateral neighbours of all database images, we generate a network that lends itself to browsing. Lateral neighbours share some properties of the focal image, but not necessarily all. For example, a lateral neighbour may share text annotations with the focal image, but no visual similarity with it at all, or it may have a very similar colour distribution, but no structural similarity, or it may be similar in all features except shape, etc. As a consequence, lateral neighbours are deemed to expose the polysemy of the focal image. Hence, when they are presented, the user may then follow one of them by making it the focal image and explore its lateral neighbours in turn. The user interaction is immediate, since the underlying network was computed offline.

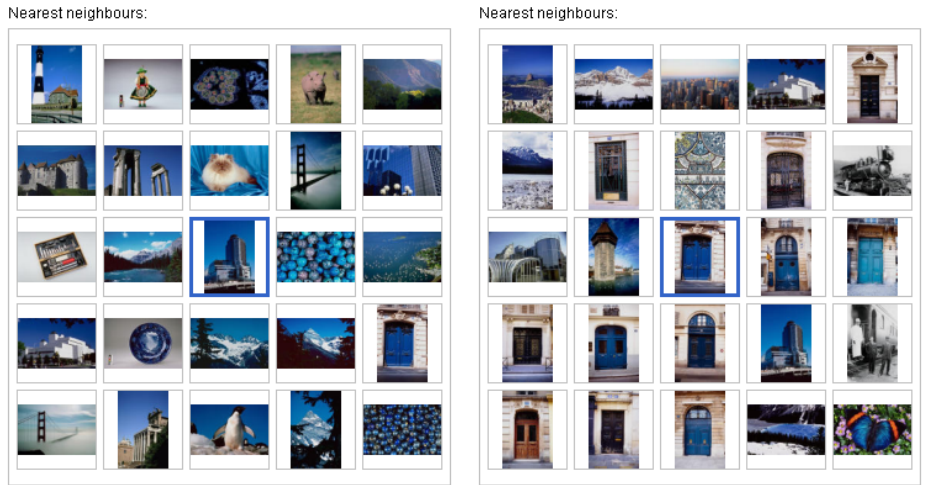
We provide the user with entry points into the database by computing a representative set of images from the collection: we cluster high-connectivity nodes and their neighbours up to a certain depth using the Markov chain clustering algorithm [75], which has robust convergence properties and allows one to specify the granularity of the clustering. The clustering result can be seen as a image database summary that shows highly-connected nodes with far-reaching connections. The right panel of Fig 7(a) is such a summary for our Corel database. The user may select any of these images as an entry point into the network. Clicking on an image moves it into the center around which the lateral neighbours are displayed, see the nearest-neighbour panel on the left side of Fig 7(a). If the size of the lateral-neighbour set is above a certain threshold the actual number of images displayed is reduced to the most salient ones.

If a user wanted to find images of blue doors then they might explore the database in Fig 7 by clicking on the blue¹ butterfly. The resulting lateral neighbours, displayed in the left panel of Fig 7(a), do not contain doors; however, they contain an image of skyscrapers which share structural similarities with doors (both have oblong structures). Indeed, making the image of the skyscraper the focal image, as seen in the left part of Fig 7(b), reveals it has a blue-door image as lateral neighbour. Clicking that will unearth a lot more images of blue doors,

¹ The online version of this article on <http://www.springeronline.com> is in colour.



(a) initial visual summary of the database (right panel) from which the user chooses the butterfly: then its nearest lateral neighbours are displayed in the left panel



(b) clicking on any image will make it the centre of the nearest neighbours panel and display is associated lateral neighbours around it

Fig. 7. Lateral browsing for a blue door

see the right side of Fig 7(b). The network structure, a bit of lateral thinking and three mouse clicks have brought the desired result.

In the same way, and with only three clicks, complicated queries such as “find video shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at” (TRECVID topic 102 [74]) can be satisfied in an image database representing 32,000 video shots, and interactive retrieval evaluations have proven this method to be effective [38,35,34]. Heesch has shown [33,37] that this is no coincidence: lateral-neighbour networks computed in this way have the so-called *small world property* [76] with only 3–4 degrees of separation even for the large TRECVID databases.

Geo-temporal browsing takes the idea of timelines and automatically generated maps, eg as offered in the Perseus Digital Library [24, 61], a step further: it integrates the idea of browsing in time and space with a selection of events through a text search box. In this way, a large newspaper or TV news collection could be made available through browsing based on what happened where and when as opposed to by keyword only.

The interface in Fig 8 was developed during a master’s project [78]; it allows navigation within a large news event dataset along three dimensions: time, location and text subsets. The search term presents a text filter, so in this case, the user is only interested in events pertaining to elections. The time interval panel defines the relevant time-interval with two sliders: The upper slider allows a coarse narrowing of the time interval, as it defines the bounds of the second slider directly underneath, which is for a finer grained selection. In this example, only events that occurred between January 1994 and December 1997 are

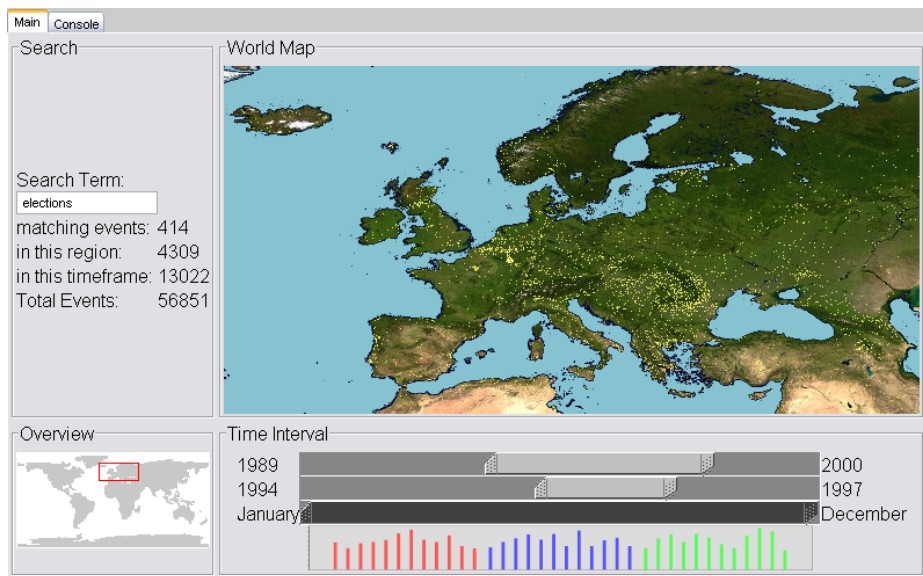


Fig. 8. Geo-temporal browsing in action

displayed. The search space can also be limited to a particular region of interest. The overview window establishes a frame of reference for the user's region of interest. Each number in the search panel counts the events within the current selection of the particular search parameter, independent of any of the other parameters. The histogram below the timeline represents the number of events within each month of the current selection. In principle, this interface could implement new zooming techniques, eg speed-dependent automatic zooming [20,19], and link to a server holding a large quantity of maps such as National Geographic's MapMachine (<http://plasma.nationalgeographic.com/mapmachine/> as of May 2005) with street-level maps and aerial photos.

7 Discussion and Future Work

This paper has given some examples of user-centred methods that support resource discovery in multimedia digital libraries. Each of these methods can be seen as an alternative mode to, and not as a replacement of, the traditional digital library management tools of meta-data and classification. The new visual modes aim at generating a multi-faceted approach to present digital content: *video summaries* as succinct versions of media that otherwise would require a high bandwidth to display and considerable time by the user to assess; *information visualisation* techniques help the user to understand a large set of documents that match a query; *visual search* and *relevance feedback* afford the user novel ways to express their information need without taking recourse to verbal descriptions that are bound to be language-specific; alternative resource discovery modes such as *lateral browsing* and *geo-temporal browsing* will allow users to explore collections using lateral associations and geographic or temporal filters rather than following strict classification schemes that seem more suitable for trained librarians than the occasional user of multimedia collections. The cost for these novel approaches will be low, as they are automated rather than human-generated. It remains to be seen how best to integrate these services into traditional digital library designs and how much added value these services will bring about. Our group has started a multimedia digital libraries project that aims to answer these open questions [6].

Acknowledgements. The paradigms outlined in this paper would not have been possible without the ingenuity, imagination and hard work of all the people I am fortunate to work with or to have worked with: Matthew Carey, Daniel Heesch, Peter Howarth, Partha Lal, João Magalhães, Alexander May, Marcus Pickering, Jonas Wolf, Lawrence Wong and Alexei Yavlinsky.

References

1. C Aggarwal and P Yu. The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *KDD*, 2000.
2. M Ankerst, D Keim and H Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *VIS*, 1996.

3. J Aslam and M Montague. Models for metasearch. In *SIGIR*, 2001.
4. J Assa, D Cohen-Or and T Milo. Displaying data in multidimensional relevance space with 2d visualization maps. In *VIS*, 1997.
5. P Au, M Carey, S Sewraz, Y Guo and S Ruger. New paradigms in information visualisation. In *SIGIR*, 2000.
6. D Bainbridge, P Browne, P Cairns, S Ruger and L-Q Xu. Managing the growth of multimedia digital content. In preparation, 2005.
7. B Bartell, G Cottrell and R Belew. Automatic combination of multiple ranked retrieval systems. In *SIGIR*, 1994.
8. J Beis and D Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR*, 1997.
9. D Blei and M Jordan. Modeling annotated data. In *SIGIR*, 2003.
10. K Borner. Visible threads: A smart VR interface to digital libraries. In *IST/SPIE*, 2000.
11. I Campbell. *The ostensive model of developing information-needs*. PhD thesis, Uni of Glasgow, 2000.
12. S Card. Visualizing retrieved information: A survey. *IEEE Computer Graphics and Applications*, 16(2):63–67, 1996.
13. M Carey, D Heesch and S Ruger. Info navigator: a visualization interface for document searching and browsing. In *DMS*, 2003.
14. G-H Cha. Bitmap indexing method for complex similarity queries with relevance feedback. In *ACM MMDB workshop*, 2003.
15. M Chalmers and P Chitson. Bead: Explorations in information visualisation. In *SIGIR*, 1992.
16. B Chawda, B Craft, P Cairns, S Ruger and D Heesch. Do "attractive things work better"? An exploration of search tool visualisations. In preparation, 2005.
17. M Christel and A Warmack. The effect of text in storyboards for video navigation. In *ICASSP*, May 2001.
18. M Christel, A Warmack, A Hauptmann and S Crosby. Adjustable filmstrips and skims as abstractions for a digital video library. In *ADL*, 1999.
19. A Cockburn, J Looser and J Savage. Around the world in seconds with speed-dependent automatic zooming. In *UIST Conf supplement*, 2003.
20. A Cockburn and J Savage. Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods. In *BCS HCI*, 2003.
21. I Cox, M Miller, T Minka, T Papathomas and P Yianilos. The Bayesian image retrieval system, PicHunter. *IEEE Trans on Image Processing*, 9(1):20–38, 2000.
22. K Cox. Information retrieval by browsing. In *Int'l Conf on New Information Technology*, 1992.
23. K Cox. *Searching through browsing*. PhD thesis, Uni of Canberra, 1995.
24. G Crane, editor. *Perseus Digital Library Project*. Tufts Uni, 30 May 2005, <http://www.perseus.tufts.edu>, 2005.
25. B Croft and T Parenty. Comparison of a network structure and a database system used for document retrieval. *Information Systems*, 10:377–390, 1985.
26. H Cunningham. GATE, a general architecture for text engineering. *Computers and the Humanities*, 36:223–254, 2002.
27. H Cunningham, D Maynard, K Bontcheva and V Tablan. GATE: A framework and graphical development environment for robust nlp tools and applications. In *ACL*, 2002.
28. A de Vries, N Mamoulis, N Nes and M Kersten. Efficient k-nn search on vertically decomposed data. In *SIGMOD*, 2002.

29. D Dearholt and R Schvaneveldt. Properties of Pathfinder networks. In R Schvaneveldt, editor, *Pathfinder associative networks: Studies in knowledge organization*. Norwood, 1990.
30. P Duygulu, K Barnard, N de Freitas and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.
31. S Feng, R Manmatha and V Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR*, 2004.
32. R Fowler, B Wilson and W Fowler. Information navigator: An information system using associative networks for display and retrieval. Technical Report NAG9-551, 92-1, Dept of Computer Science, University of Texas, 1992.
33. D Heesch. *The NN^k technique for image searching and browsing*. PhD thesis, Imperial College London, 2005.
34. D Heesch, P Howarth, J Magalhães, A May, M Pickering, A Yavlinsky and S Rüger. Video retrieval using search and browsing. In *TRECVID*, 2004.
35. D Heesch, M Pickering, S Rüger and A Yavlinsky. Video retrieval using search and browsing with key frames. In *TRECVID*, 2003.
36. D Heesch and S Rüger. Performance boosting with three mouse clicks — relevance feedback for CBIR. In *ECIR*, 2003.
37. D Heesch and S Rüger. NN^k networks for content based image retrieval. In *ECIR*, 2004.
38. D Heesch and S Rüger. Three interfaces for content-based access to image collections. In *CIVR*, 2004.
39. D Heesch and S Rüger. Query-based keyword extraction and document clustering for information retrieval and knowledge consolidation. In preparation, 2005.
40. D Heesch, A Yavlinsky and S Rüger. Performance comparison between different similarity models for CBIR with relevance feedback. In *CIVR*, 2003.
41. M Hemmje, C Kunkel and A Willet. Lyberworld — a visualization user interface supporting fulltext retrieval. In *SIGIR*, 1994.
42. P Hoffman, G Grinstein and D Pinkney. Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. In *NPIV 1999*, 2000.
43. P Howarth and S Rüger. Trading precision for speed: localised similarity functions. In *CIVR*, 2005.
44. Y Ishikawa, R Subramanya and C Faloutsos. MindReader: Querying databases through multiple examples. In *VLDB*, 1998.
45. E Izquierdo and D Djordjevic. Using relevance feedback to bridge the semantic gap. In *AMR*, 2005.
46. J Jeon and V Lavrenko and R Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR*, 2003.
47. R Korfhage. To see or not to see — is that the query? In *SIGIR*, 1991.
48. V Lavrenko, R Manmatha and J Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.
49. A Lelescu, O Wolfson and B Xu. Approximate retrieval from multimedia databases using relevance feedback. In *SPIRE/CRIWG*, 1999.
50. C Meilhac and C Nastar. Relevance feedback and category search in image databases. In *ICMCS*, 1999.
51. D Metzler and R Manmatha. An inference network approach to image retrieval. In *CIVR*, 2004.
52. Y Mori, H Takahashi and R Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Int'l Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

53. W Müller and A Henrich. Faster exact histogram intersection on large data collections using inverted VA-files. In *CIVR*, 2004.
54. S Nene and S Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Trans Pattern Anal Mach Intell*, 19(9):989–1003, 1997.
55. L Nowell, R France, D Hix, L Heath and E Fox. Visualizing search results: Some alternatives to query-document similarity. In *SIGIR*, 1996.
56. M Pickering. *Video Retrieval and Summarisation*. PhD thesis, Imperial College London, 2004.
57. M Pickering, L Wong and S Rüger. ANSES: Summarisation of news video. In *CIKM*, 2003.
58. K Porkaew, M Ortega and S Mehrotra. Query reformulation for content based multimedia retrieval in MARS. In *ICMCS*, 1999.
59. K Rodden, W Basalaj, D Sinclair and K Wood. Evaluating a visualization of image similarity. In *SIGIR*, 1999.
60. Y Rui, T Huang and S Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *SPIE*, 1998.
61. J Rydberg-Cox, R Chavez, A Mahoney, D Smith and G Crane. Knowledge management in the perseus digital library. *Ariadne*, 2000.
62. J Rydberg-Cox, L Vetter, S Rüger and D Heesch. Approaching the problem of multi-lingual information retrieval and visualization in greek and latin and old norse texts. In *ECDL*, 2004.
63. J Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans on Computers*, C-18(5), 1969.
64. S Santini, A Gupta and R Jain. Emergent semantics through interaction in image databases. *IEEE Trans on Knowledge and Data Engineering*, 13(3):337–351, 2001.
65. S Santini and R Jain. Integrated browsing and querying for image databases. *IEEE Multimedia*, 7(3):26–39, 2000.
66. C Shaw, J Kukla, I Soboroff, D Ebert, C Nicholas, A Zwa, E Miller and D Roberts. Interactive volumetric information visualization for document corpus management. *Digital Libraries*, 2(2/3):144–156, 1999.
67. J Shaw and E Fox. Combination of multiple searches. In *TREC 3*, 1994.
68. B Shneiderman, D Feldman, A Rose and X Ferre’ Grau. Visualizing digital library search results with categorical and hierarchical axes. In *ACM Digital Libraries*, 2000.
69. A Smeaton, C Gurrin, H Lee, K Mc Donald, N Murphy, N O’Connor, D O’Sullivan, B Smyth and D Wilson. The Físchlár-news-stories system: Personalised access to an archive of TV news. In *RIAO*, 2004.
70. A Spoerri. InfoCrystal: A visual tool for information retrieval & management. In *CIKM*, 1993.
71. D Squire, W Müller, H Müller and T Pun. Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters*, 21(13–14):1193–1198, 2000.
72. A Torralba and A Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14:391–412, 2003.
73. R Torres, C Silva, C Medeiros and H Rocha. Visual structures for image browsing. In *CIKM*, 2003.
74. TRECVID. Trec video retrieval evaluation. <http://www-nlpir.nist.gov/projects/trecvid/>, 2005.
75. S van Dongen. A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

76. D Watts and S Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
77. R Weber, H-J Stock and S Blott. A quantitative analysis and performance study for similarity search methods in high-dimensional space. In *VLDB*, 1998.
78. J Wolf. GeoBrowser: A graphical approach to geo-temporal news browsing. Master's thesis, Imperial College London, 2005.
79. M Wood, N Campbell and B Thomas. Iterative refinement by relevance feedback in content-based digital image retrieval. In *ACM Multimedia*, 1998.
80. A Yavlinsky, M Pickering, D Heesch and S Ruger. A comparative study of evidence combination strategies. In *ICASSP*, 2004.
81. A Yavlinsky, E Schofield and S Ruger. Automated image annotation using global features and robust nonparametric density estimation. In *CIVR*, 2005.
82. M Zhou and S Feiner. Visual task characterization for automated visual discourse synthesis. In *CHI*, 1998.

Using Relevance Feedback to Bridge the Semantic Gap

Ebroul Izquierdo and Divna Djordjevic

Queen Mary University of London, MileEnd Road, E1 4NS, London, UK
{ebroul.izquierdo, divna.djordjevic}@elec.qmul.ac.uk

Abstract. In this article relevant developments in relevance feedback based image annotation and retrieval are reported. A new approach to infer semantic concepts representing meaningful objects in images is also described. The proposed technique combines user relevance feedback and underlying low-level properties of elementary building blocks making up semantic objects in images. Images are regarded as mosaics made of small building blocks featuring good representations of colour, texture and edgeness. The approach is based on accurate classification of these building blocks. Once this has been achieved, a signature for the object of concern is built. It is expected that this signature features a high discrimination power and consequently it becomes very suitable to find other images containing the same semantic object. The model combines fuzzy clustering and relevance feedback in the training stage, and uses fuzzy support vector machines in the generalization stage.

1 Introduction

The creation rate of new technologies seems to be proportional to our ability to retrieve and use specific and accurate information extracted from the collective expertise building the universe of data that supports the modern technological society. However, the exponential growth of written digital data and other forms of content including audiovisual assets, along with the critical lack of tools to record that data in a well-structured form, is rendering useless vast portions of available information. Clearly, information is worthless if it cannot be found and used. As a consequence information retrieval technology is destined to become pervasive in almost every aspect of daily life and a pillar for key-achievements in future scientific and technologic developments. Although the fundamentals of information retrieval were laid many years ago, this was done for text databases and most of current well-established retrieval tools are only suitable for text mining. Compared with text-based information retrieval, image and video retrieval is not only less advanced but also more challenging. Though writing was explicitly developed to share and preserve information while pictures and sound have been traditionally used to express human's artistic and creative capacity, this tendency is shifting in the digital age. This shifting is revolutionizing the way people process and look for information, from "text only" to "multimedia-based" search and retrieval. This change is a consequence of the rapid growth in consumer-oriented electronic technologies, e.g. digital cameras, camcorders and mobile phones, along with the expansion and globalization of networking facilities. Indeed, the availability of a wide range of digital recorders accessible to anyone, from cheap digital cameras to complex professional movie capturing devices, is

enabling the wide use of images, diagrams and other audiovisual means to record information and knowledge while boosting the content growth in digital libraries. The immediate consequence of this trend is that generating digital content has become easy and cheap while managing and structuring it to produce effective services has not. This applies to the whole range of content owners, from professional digital libraries with their terabytes of visual content to private collectors of digital pictures stored in disks of conventional personal computers.

In order to get closer to the vision of useful multimedia-based search and retrieval, the annotation and search technologies need to be efficient and use semantic concepts that are natural to the user. Finding a specific image of the last holidays stored some where in the hard disk of the personal computer can become a time consuming task, searching for a specific picture in a large digital archive is even more difficult, and looking for an image on the web can be extremely difficult if not impossible. If images have been manually labelled with an identifying string, e.g. "My holidays in Africa, wild life", then the problem may appear to have been finessed. However, the adequacy of such a solution depends on human interaction, which is expensive, time consuming and therefore infeasible for many applications. Even the annotation of few hundreds of personal images captured during the last years is a tedious task that nobody wants to do. Furthermore, such semantic based annotation is completely subjective and depends on semantic accuracy in describing the images. While one person could label an image as "wild life" someone else might prefer "Elephant and trees". This last mentioned problem could be alleviated by constraining the annotation to words extracted from a pre-defined dictionary, taxonomy or advanced ontology. Nevertheless, the challenge of automatic annotation and retrieval using semantic structures natural to humans remains critical. "*The holy grail of content based image retrieval (CBIR) research is to bridge the semantic gap between the current capabilities of CBIR and the needs of users (or how they expect to be able to search, e.g., via concepts or emotions)*" [1].

To develop the technology able to produce accurate levels of abstraction in order to annotate and retrieve content using queries that are natural to humans is the breakthrough needed to bridge the semantic gap. In this article the semantic gap in multimedia processing is defined as *the discrepancy between low-level features or content descriptors that can be computed automatically with current algorithms, and the richness of semantics in user queries due to the subjectivity of high-level human interpretations of audiovisual media*. To bridge this gap is a challenge that has captured the attention of researchers in computer vision, pattern recognition, image processing and other related fields, evidencing the difficulty and importance of such technology and the fact that the problem is unsolved. The underlying technology offers the possibility of adding the audiovisual dimension to well-established text databases enabling multimedia based information retrieval.

Much related work on image indexing and retrieval has focused on the definition of low-level descriptors and the generation of metrics in the descriptor space. These techniques are aimed at defining image signatures using primitives extracted from the content patterns, e.g., pixel patterns and dynamics in image and video or sampling patterns in audio signals. These signatures are called low-level descriptors and represent information that can be extracted automatically and directly from the content. Although low-level descriptors are extremely useful when a query by example is

considered, they have little in common with high-level semantic concepts. Query by example uses similarity metrics acting on low-level features such as colour, texture, shape, motion, and audio primitives. It is based on the assumption that the user has an example, e.g., picture, video clip or song. There is a clear analogy between query by example based image retrieval and text retrieval: both assume that the query has the same structure as the database content. Text retrieval searches for patterns of interest and similarities in text databases using text as query, query by example in images also searches for patterns of interest and similarities in a meta-database containing low-level descriptions of the image database, using low-level descriptors as a query. However, if the aim is to retrieve audiovisual content using semantic structures, e.g., words or sentences, which are natural to humans, two profound challenges become evident: how to deal with the subjective interpretation of images by different users under different conditions; and how to link a semantic-based query with low-level metadata. The first problem originates in the fact that perceptual similarity is user and context dependent. The second challenge is a synonym of the semantic gap. To provide a personalized, user tailored, result the machine needs to learn user preferences and inclinations and to discover low-level pattern representation of these preferences. To bridge the semantic gap the machine needs to learn associations between low-level patterns and semantic concepts. In either case the need for learning and inference technology involving user's input becomes evident.

Early image annotation and retrieval methods are based on conventional machine learning and pattern recognition, e.g., clustering analysis. The idea behind these methods is to infer semantic classes from low-level descriptors using well-established recognition techniques [2]. These techniques use pattern recognition on manually labelled content to train the system. The learning process is based on basic visual interpretation of the image content indicating observed elements in the scene, e.g. landscape, cityscape [3]. The aim is to link visual primitives with few semantic classes and to use these few generic high-level descriptions to annotate the images in the database. This bottom-up approach, from low-level to semantic classification, mostly relies on matching procedures and pattern recognition at the lowest level of content interpretation. Since two objects can have similar low-level primitives while being semantically different to a human observer, substantial noise in the annotation process is introduced when a small, annotated database is used to propagate semantic labels over large databases. The same occurs when the number of classes, i.e., the annotation granularity, is increased from few to many semantic concepts. Other techniques start at the other side of the semantic gap. Using well defined ontologies and few manually annotated images, the aim is to propagate "words" to the whole database using relationships and rules defined over the underlying ontology. This top-down approach puts a heavy burden on the designer of the high-level relations. Furthermore, it only delivers satisfactory results for specific application scenarios where a limited and well-defined ontology can be constructed. Combined approaches aimed at closing the gap from both directions, i.e., top-down and bottom-up simultaneously, appear more promising. However, they are also limited to specific application scenarios where ontological structures make sense [4].

The main drawback of all these methods is that they do not consider the subjectivity of user's interpretations. Moreover, their efficiency decreases proportional to the number of scenarios and application cases. Consequently, to provide personalized

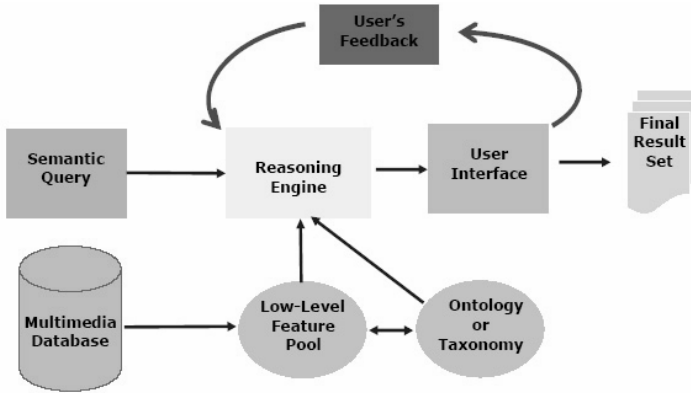


Fig. 1. Generic architecture of annotation and retrieval system with relevance feedback

retrieval performance considering user preferences while improving semantic annotation, some degree of user input is needed. The most natural way of getting user's subjective information and preferences into the system is by using models that incorporate the following two user centred features: online learning from the user interactions with search engine; and low-complexity user friendly interfaces to enter user's judgments on retrieved results. The idea underpinning this model is to integrate a "relevance feedback" loop into the system with the user at the centre and the machine learning from user's feedback. Clearly, relevance feedback (RF) schemes enable improved search performance using online learning strategies. The RF concept is based on the analysis of relevant and irrelevant information fed back into the system by the user. This analysis predicts and learns user's preferences in order to iteratively improve retrieval results. Semi-automatic adaptive learning strategies based on relevance feedback are aimed at learning relations between high-level semantic concepts used by humans to identify objects in an image and low-level descriptions of the same visual information. Fig. 1 depicts the overall architecture of a retrieval engine featuring relevance feedback.

The objective of this article is to describe relevant developments in RF-driven image annotation and retrieval and to introduce a new model to infer semantic concepts for the annotation of meaningful objects in an image. The model exploits underlying low-level properties of elementary image blocks that make up images. An important feature of this approach is that the emphasis is on single objects rather than on the whole scene depicted in the image. The proposed technique was inspired by three observations: users are mostly interested in finding objects in images and do not care about the surroundings in the picture, e.g. background or other objects; elementary image blocks are closer to low-level descriptions than whole objects or images; and objects can be regarded as mosaics made up these building blocks. The approach combines fuzzy support vector machine and RF in an intuitive framework for semi-automatic image annotation.

The remaining of the paper is organized as follows. Section 2 presents an overview of the most prominent classes of models for image annotation and retrieval using RF. Section 3 describes the fundamentals of support vector machines (SVM) as learning and classification schema and fuzzy SVMs used in the proposed framework. Section 4

describes selected results of computer experiments using SVMs and clustering methods. The framework for semiautomatic annotation and retrieval using elementary image blocks is outlined in section 5. The paper closes with conclusions in section 6.

2 RF Based Image Retrieval and Annotation Systems

Relevance Feedback was originally developed for retrieval on text databases. After few years, the concept was adopted for the more challenging problem of audiovisual information retrieval. RF is an essential feature of advanced visual-based retrieval and annotation systems and probably key to close the semantic gap.

Several RF algorithms have been proposed over the last few years. The idea underpinning most RF-models is that the distance between images labelled as relevant and other similar images in the database should become minimal. Though the human visual system does not obey a specific mathematic metric when looking for similarity between pictures, the distances used in image retrieval systems follow well-defined metrics in a feature space. As a consequence different aspects need to be considered when modelling the system: how to define the feature space as a representation of the image database and reflecting the properties of the human visual system; how to select features in the feature space; and how to define distances between vectors in the feature space. RF-based image retrieval systems can be classified according to the underlying feature space, or the generic approach used to define metrics in feature space, and most importantly according to the process used to learn and link low-level features with semantic concepts derived or inferred from user preferences and notion of similarity between images. The most prominent models are outlined in the sequel.

2.1 Descriptive Classification Models

These models exploit RF in a form of relative judgment. The PicHunter system described in [5] uses “stochastic-comparison search”. Once a user selects relevant images, a comparison search is done over the whole database. In the simplest one-dimensional case, a binary search comparing any element from the meta-database and the target vector is conducted. In a more general case, “the vantage-point tree algorithm”, as version of a KD-tree approach is used. Other early models use feature re-weighting or query point movement strategies [6], [7], [8]. Feature re-weighting is based on “term frequency” and “inverse document frequency” techniques similar to those studied and used in text retrieval. The idea behind query point movement techniques is to move the query feature vector towards positive examples and away from negative examples. This approach is based on the assumption that all positive examples have similar feature vectors and can be cluster together in feature space. Heuristic techniques based on feature re-weighting and query point movement use empiric parameter adaptation annealing well-established weighting methods in text-based retrieval. In this model, features are weighted differently according to their classification power. The classification power is derived from the analysis of relevant images fed back by the user. Features providing the most compact clustering of relevant images and separation of relevant and irrelevant image get a stronger weight in the classification process. Techniques based on probabilistic models have been also widely studied. Early probabilistic RF systems assumed that positive images follow a

single Gaussian distribution [8]. More advanced systems are based on Gaussian mixture models. Though this assumption does not offer a generic solution, it is more sensible, since semantic concepts having similar meaning are usually widely distributed in the feature space. To estimate the parameters of the Gaussian mixture it is assumed that positive examples can be grouped in feature space and that they are mutually separated by negative examples.

2.2 Neural Networks and Relevance Feedback

A large number of techniques for image retrieval integrate neural network learning approaches and relevance feedback. Among different strategies, self-organizing maps are popular to index images according to low-level features. Self-organizing maps are unsupervised topologically ordered neural networks, which project a high-dimensional input space into a low-dimensional lattice. The latter, usually being a two-dimensional grid with n -dimensional neighbours connected in appropriately weighted nodes. Some schemas, e.g., the PicSOM system [9], reduce the complexity inherent to the training of large self-organizing maps using a hierarchical structure. It tackles complexity organizing similar feature vectors into neighbouring neurons, so that relevance information is mapped from the images labelled by the user to appropriate best-matching map units. Another class of approaches combines neural network based learning with fuzzy RF in the user iteration process [10]. In this model the user provides a fuzzy judgment about the relevance of an image. This strategy contrasts binary RF systems in which a hard relevance decision is used. In a first iterative step, a hierarchical tree with multiple levels of information provided to the user is defined. The user is requested to label images as relevant, irrelevant, or fuzzy when he/she is unsure about the degree of relevance. A continuous fuzzy membership function models user's fuzzy feedback. In response to user's perception images are weighted with different fuzzy factors. The learning process involves user's preferences and visual content interpretation combined in a single neural network.

2.3 Discriminative Analysis

In some cases RF leads to a two-class problem, i.e., relevant, irrelevant. Thus, Discriminative Analysis can be used to tackle the underlying classification problem. In the linear case Fisher's Discriminate can be applied [11]. The aim is to find linear projections such that classes are well separated. Separability is measured by how far the projected means of two classes are apart and how large is the variance of the data along the projected direction. Discriminative Expectation Maximization considers image retrieval as a learning problem with labelled and unlabeled data samples [11]. It is used to estimate both, the parameters of the probabilistic density model and the linear transformation that maps the original feature space into another feature space. In the general case when the number of components in the mixture model is unknown, Expectation Maximization fails to give an effective probabilistic model. Therefore a mapping that clusters original data into a new feature space, with the probabilistic density captured by Gaussian mixtures is sought. This is achieved by multiple discriminative analyses. The aim is to find a linear transformation based on the labelled data set and generalize it to the unlabelled dataset so that the inter-class scattering is maximized and intra-class scattering minimized.

The main cause for the failing of techniques based on multiple discriminative analyses is the assumptions that both positive and negative samples form coherent clusters according to the used probabilistic model. This drawback is tackled by several authors using Biased Discriminative Analysis [11], [12]. The approach is based on the assumption that all the positive examples are clustered in one class in a non-linear way, but negative examples are not clustered at all since they belong to many different classes. In this case a biased classification problem is defined assuming $n+1$ unknown classes and with the user being interested in the positive class only. The goal is to determine a function that enables maximum clustering or minimum scattering of the positive examples and maximal scattering of negative examples embraced by the n unknown classes moving them away from the positive class.

3 Support Vector Machines and Relevance Feedback

Over the last few years there has been a significant interest in the integration of SVM and relevance feedback for supervised learning. SVM are optimal hyperplane classifiers acting over a well-defined dot product feature space X . The elements of the feature space are the training patterns x_1, \dots, x_m , where each x is an N -dimensional feature vector. For binary classification a training data set generated by an unknown probability distribution $P(x)$ is used. For each pattern the supervised input y is derived from the unknown conditional distribution $P(y|x)$.

$$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m) \in \mathbf{R}^N \times \{-1, +1\} \quad (1)$$

Starting from a set of functions $f(x)$ mapping $f: \mathbf{R}^N \rightarrow \{+1, -1\}$, the aim is to estimate the optimal classification function for an expected risk on the training dataset. The margin is the minimal distance from samples to the decision surface. The decision surface is called optimal hyperplane. The class of hyperplanes is given as:

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0, \mathbf{w} \in \mathbf{R}^N, b \in \mathbf{R} \quad (2)$$

The corresponding decision function classifier for correctly predicting label y_i , is given as:

$$f(x) = \text{sgn}((\mathbf{w} \cdot x) + b) \quad (3)$$

The hyperplane is optimal if it separates a set of vectors and at the same time maximizes the distance between the vectors which are closest to the hyperplane. It is well-known that the optimal hyperplane is orthogonal and on the halfway of the shortest line connecting the convex hulls of the two classes. There is also a logical connection between SVM, structural risk minimization and Vapnik-Chervonenkis (VC) dimension [13], [14], [15]. If vectors $x \in X$, x_1, \dots, x_m belong to a sphere of radius R enclosing all the data $x \in X$, $\|x - a\| < R$, where $a \in X$ is the centre of the sphere, then for a set of hyperplane functions obeying $\|\mathbf{w}\| < A$, the VC-dimension satisfies:

$$h \leq R^2 A^2 + 1 \quad (4)$$

Moreover, the margin of a hyperplane has a lower bound, with the distance from any sample to hyperplane satisfying:

$$d(\mathbf{w}, b, x) \geq \frac{1}{A} \tag{5}$$

Thus, for a large lower bound, the VC-dimension is small, and vice versa, with a small margin larger class of problems can be separated, Fig. 2. Only a hyperplane that is further from any sample than $1/A$ can be a potential optimal hyperplane. Consequently the number of possible planes is decreased, and the capacity as the margin is increased.

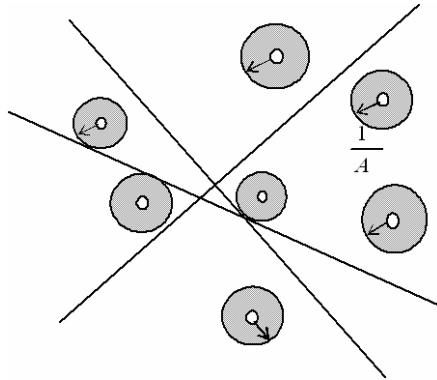


Fig. 2. Constraints on canonical hyperplanes

Generic SVM are not very suitable for applications where the input data is fuzzy. To deal with fuzziness Lin and Wang in [16] enable incorporation of membership values for each sample with relative importance, by penalizing the slack variable and therefore decreasing allowed distance from the margin. The resulting SVM features as many free parameters as the number of training samples. In [17] the similar formulation of SVM is considered, fuzzy prior knowledge is incorporated with each training sample associated with a confidence value.

These specific SVM are termed Fuzzy SVM (FSVM). FSVM are also defined in the same dot product feature space X as classical SVMs. However, it is assumed that each pattern has a level of relevance or membership to a particular class:

$$(x_1, y_1, u_1), (x_2, y_2, u_2), \dots, (x_m, y_m, u_m) \in \mathbf{R}^N \times \{-1, +1\} \times [\delta, 1] \tag{6}$$

In (6) δ is greater than zero, since zero membership is equivalent to having no additional information. Solving the resulting optimisation problem (7) is equivalent to maximizing the margin and minimizing the overall error measure with variable values according to the fuzzy relevance of each pattern. This procedure leads to the estimation of the optimal hyperplane (3):

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m u_i \xi_i \tag{7}$$

$$y_i((\mathbf{w} \cdot x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \dots, m \quad (8)$$

Furthermore, minimizing the variable error measure $\sum u_i \xi_i$, weighted according to the relevance of the samples, leads to a minimization of the number of misclassified training samples and the empirical error. Here C controls both the margin and the amount of misclassified errors. Thus, minimizing (7) with a larger value of C results in less misclassification since the margin is decreased. The solution of (7) is achieved by converting the primal problem into a dual problem and by solving the later one. Then the Lagrangian coefficients α_i different from zero define the support vectors, which lie on the optimal hyperplane and define the decision function:

$$f(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i \cdot (x \cdot x_i) + b\right). \quad (9)$$

This model can be used in the prediction process to generate labels for the testing set. For the non-linear case the inner product in (1-9) is replaced by a kernel function. This kernel is a non-linear mapping of the input space into a higher dimensional feature space. It may enable a linear separation of otherwise non-separable data.

4 Experimental Evaluation of RF Based on SVM and FSVM

To evaluate the performance of combined relevance feedback and conventional SVM, as well as FSVM, several experiments were conducted. The used test database was the Corel stock gallery. Two groups consisting of 1035 and 1200 photographs were organized into a number of semantic categories. The first group was used to classify

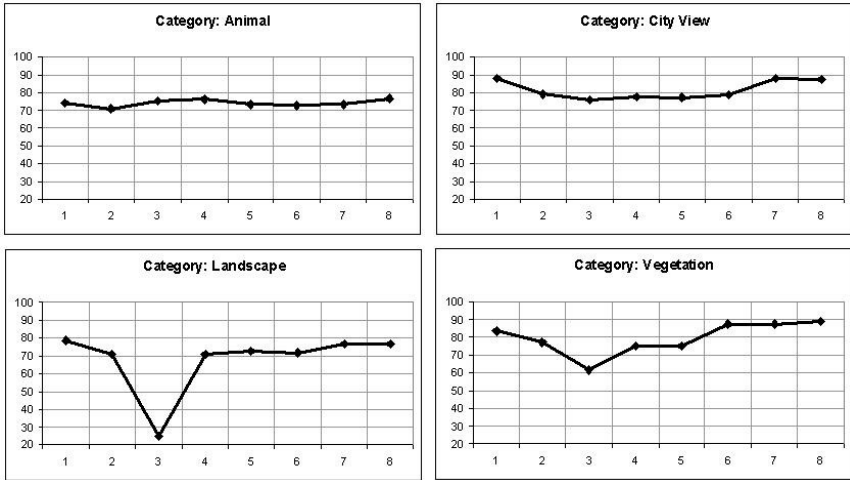


Fig. 3. Classification results using random selection of images and SVM. The X-axis indicates the number of RF-iterations. The Y-axis shows the resulting accuracy in 100%.

animals, city views, and vegetation. The second group was used for indoor and outdoor images. The animal/city view/landscape/vegetation feature space combines MPEG-7 descriptors, colour structure, edge histogram, and homogeneous texture descriptors. The indoor/outdoor feature space was built with vectors containing colour layout descriptors. The training data sets were randomly generated using a fraction of the images. The remaining images were used for testing the classifier model. In order to evaluate the stability of the classifier, a set of experiments were carried out using random selections of samples [18]. As shown in Fig. 3, the classifier based on conventional SVM appears to be highly unstable. Mainly because user feedback is based upon visual inspection along with subjective criteria of the annotator without taking into account any low-level similarity. In contrast, clustering analysis can assist in the sample selection, and also contribute to the system's stability.

The following three approaches are used to assess the performance of the modelled classifier [18]:

- (SVM+FCM): The SVM classifier is trained with hints provided by a professional annotator using previously clustered images. The professional annotator only indicates the class label of each cluster. This lightens the burden of annotation but introduces noise and less accuracy in the results.
- (SVM+RL): SVM classifier using only RF. The classifier is trained with hints provided by a professional annotator on relevance of individual images. The selection of samples relies completely on the user assessment ignoring important relationships between low-level descriptors.
- (SVM+FCM+RL): The SVM classifier is trained combining both clustering results and RF. The observed accuracy is higher. This approach has the advantage of taking into account the underlying low-level structures as revealed by the clusters. It minimizes the required supervision and partially exploits the semantic information provided from the professional annotator.

Fig. 4 depicts a summary of results obtained for these three approaches applied to the indoor/outdoor classification task. It can be observed that the lowest accuracy is obtained when the SVM learns from clustering outcomes only. The classifier behaves

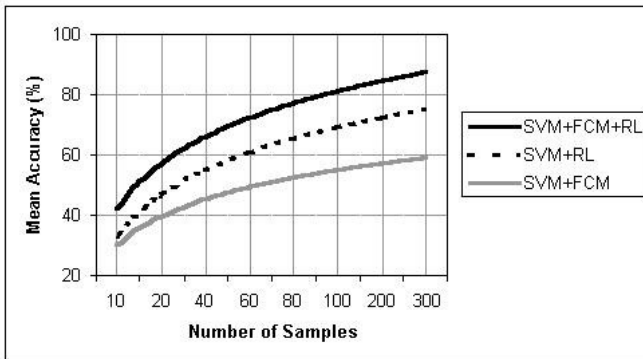


Fig. 4. Summary of results for the indoor/outdoor classification problem

better when RF and SVM are combined. The best results are obtained by using fuzzy clustering on top of combined RF and SVM. Using clusters mechanisms not only assist in the sample selection, but also contribute to the system's stability.

5 Using Elementary Building Blocks to Narrow the Semantic Gap

Most annotation and retrieval approaches from the literature have dealt with either whole images or regions segmented according to colour similarity. However, user are mostly interested in finding single semantically meaningful objects regardless of other elements that make up the whole scene. If segmentation is used the presence of noisy regions and over segmentations is unavoidable. This leads to inadequate retrieval results. Consequently, on one hand we need to consider the fact that even the best image segmentation techniques cannot extract meaningful semantic objects and it is no reasonable to assume object segmentation in a retrieval system. On the other hand, without segmentation most learning machines fail to capture the specific concept of objects the user is interested in. Indeed, neural networks, kernel machines, statistical, probabilistic algorithms and others can be trained to deliver satisfactory results for specific content classes and very constrained scenarios. Clearly, if the structure of the database is known and descriptors with high discrimination power tailored to the content are used, the underlying pattern recognition problem can be solved with relatively high accuracy. However, in a most generic case where no specific and controlled settings can be assumed the problem remains open.

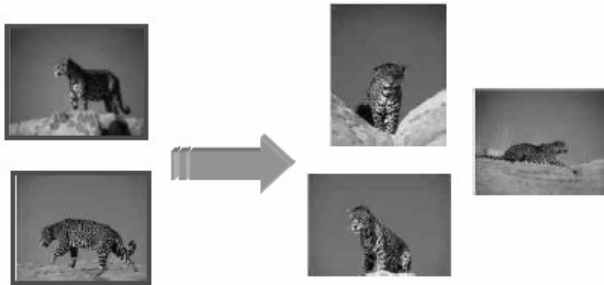


Fig. 5. Results of RF using samples of images containing the object 'jaguar' and similar background. The results at the right side are obtained using SVMs, colour and texture descriptors.

According to these considerations two different user cases can be distinguished:

- Scene retrieval assuming that the whole image depicts a semantic concept, e.g sunset, sky, landscape.
- Object oriented scenario in which the user search for semantic objects in images.

The first case exploits the fact that learning machines understand patterns represented by numerical features and these features give good semantic representations only in very specific contexts. For instance, using colour and texture as low level features and assuming that the training dataset contains the best representations of the concept in question, e.g., no occlusions, single objects in the foreground and low

variance in the background. As shown in Fig. 5, in this case the results are very promising. Though this case is easier and better understood, it is also less relevant for practical applications.

Looking at the more realistic scenario described in the second case, i.e., images containing the same object but different backgrounds as in Fig. 6, it can be observed that the learning method becomes unreliable. Here the patterns of interest are embedded in signatures full of spurious patterns that mislead the learning process. In this experiment as the number of RF-iterations increases the performance of the system decreases. This behaviour contradicts the basic concept of machine learning. The conclusion is that the learning is vastly influenced by the variety of patterns coming from background texture and other objects. Moreover, each new RF-iteration has the potential to introduce new patterns that mislead the learning process.

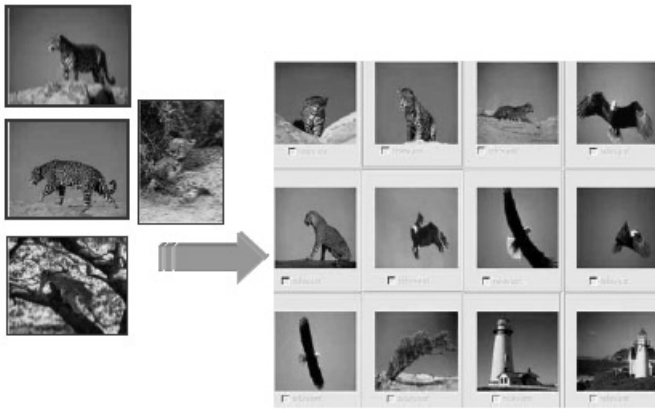


Fig. 6. RF results using the object ‘jaguar’ in front of natural backgrounds. The results at the right side are obtained using SVMs, colour and texture descriptors.

If in an experimental setting we reverse the problem and try to force the human to think as the machine by providing low-level descriptor rather than images, the learning process becomes more transparent. A detailed study and experimental evaluation of this idea is given in [18]. Following the results of these experiments we have devised a strategy to link meaningful low-level representations with semantic objects in images. Semantic objects and complete images can be regarded as mosaics of small building blocks. In most cases these building blocks do not represent semantic concepts. However, they display good sample representations of colour texture and edgeness. Thus, small picture building blocks can be regarded as being closer to low-level descriptions than to high semantic concepts. If we can classify the image building blocks accurately, the semantic annotation and retrieval problem appear to be finessed. Thus, it becomes natural to approach the problem using these building blocks as starting point. Fig. 7 shows an image containing the object ‘Elephant’. The image is subdivided into small blocks of regular size. User RF will provide selected pictures with Elephants. Now the task at hand is the classification of all blocks relevant to the object Elephant and to filter out background blocks. Once this has been achieved, a signature for the object Elephant can be built. It is expected that this signature features a high

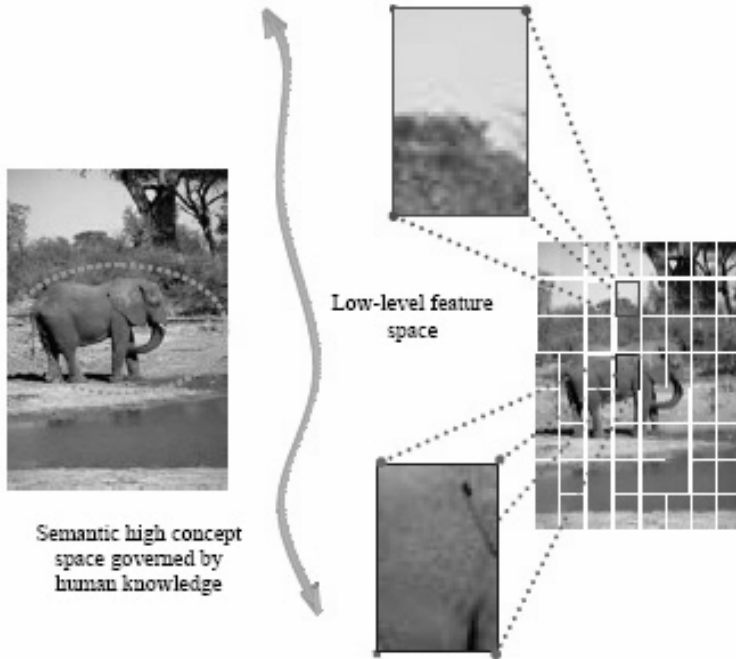


Fig. 7. Building blocks of the object or concept “elephant” (right), low-level features are closer to these elementary building blocks than to the semantic concept in the picture at the left

discrimination power and thus it is highly suitable to annotate or retrieve other images containing Elephants. For the sake of conciseness details of this approach will not be presented. In the remaining of this section a brief outline of the frameworks is given.

In the proposed approach key patterns common to all of the data samples representing an average signature for the object of interest are sought. It is assumed that the user provides fuzzy information on the relevance of an image with respect to the semantic object of concern. Intuitively all images labelled as relevant will contain a set of blocks with common low-level descriptors throughout the whole training set. Background blocks will be different to the conceptual object and they will not feature common patterns and thus clustering capabilities over the whole training set.

The first processing step consists of automatic fuzzy clustering of building blocks. Here it is assumed that the RF provides a degree of relevance (fuzziness) or how relevant is the selected image to the concept or object of concern. After clustering we two types of clusters are distinguished:

- Clusters that have a lot of blocks belonging to all or most of the images labelled as relevant. Here it is assumed that blocks making up the same semantic object will cluster together and feature a high degree of membership to that particular cluster. These clusters should have a large cardinality and their elements should be spread over the entire training set. The low-level signature extracted from these blocks is used as positive input for the FSVM. The degree of relevance (fuzziness) corresponding to the block is carried forward to the FSVM. Fig. 8 outlines blocks clustered together from a sequence of images with the object Elephant.

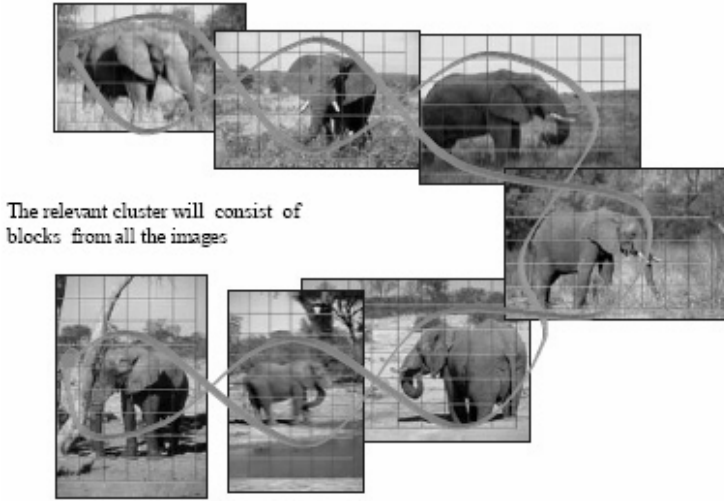


Fig. 8. Blocks representing concepts of interest are present in the relevant cluster. The cluster consists of relevant blocks from all images in the training dataset.

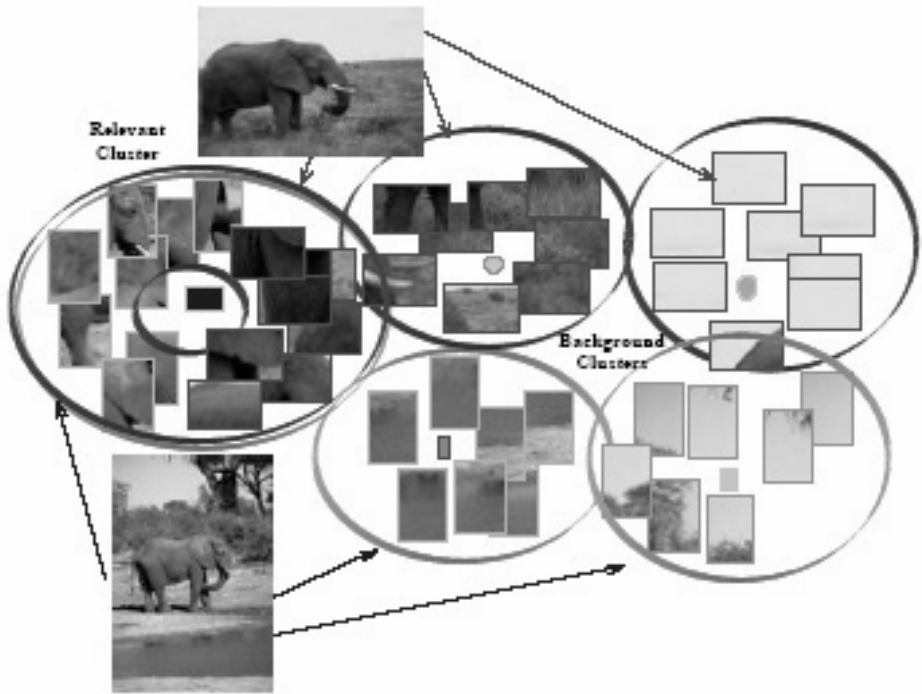


Fig. 9. Cluster structure with the relevant representative cluster and background clusters

- The second type of clusters comprises blocks coming from few images. They represent the background content not relevant to the user but still part of images labelled as relevant. These clusters are used as negative samples. Since the user has given a fuzzy relevance value to the image these blocks get assigned a negative label. The intuition behind this strategy is that positive blocks cluster close together while negative blocks from the remaining clusters are distributed over the whole feature space and for that reason their negative relevance becomes more difficult to define.

In a subsequent step, representative block features are input to a learning method based on fuzzy Support Vector Machines. Since conventional SVM requires two classes to make a distinction between relevant and irrelevant concepts, background blocks and other negative information are used as second class. Aiming at discriminating irrelevant background and non-relevant objects, less absolute relevance is given to negative samples. The results of the FSVM prediction phase are those blocks furthest to the separating hyperplane. These blocks indicate which images are returned to the user for the following RF-iteration. A selected example of block-features input to the FSVM is shown in Fig. 9.

6 Conclusions

In this article relevant developments in RF-driven image annotation and retrieval are presented. A new model for the inference of semantic concepts representing meaningful objects in images is described. The model uses FSVM to generalize classification using low-level signatures extracted from semantic objects in images. The model exploits underlying low-level properties of elementary building blocks of a semantic concept. Images are regarded as mosaics of small building blocks displaying good sample representations of colour texture and edgeness. Thus, small image building blocks are closer to low-level descriptions than to high semantic concepts. The presented approach is based on the classification of these image building blocks accurately using fuzzy clustering. Once this has been achieved, a signature for the object of concern is built. It is expected that this signature features a high discrimination power and thus it is highly suitable to annotate or retrieve other images containing the same object.

Acknowledgements

This research was partially supported by the European Commission under contract FP6-001765 aceMedia.

References

1. O'Reilly, J.: Content Engineering. *Electronics Communications Engineering Journal*, Vol. 14, No. 4, Aug. 2002
2. Vailaya, A., Figueiredo, M. A. T., Jain, A. K., and Zhang, H.-J. Image classification for content-based indexing. *IEEE Trans. Image Processing*, 2001, 10, pp. 117-130

3. Vailaya, A., Jain, A. and Zhang, H.-J.: On image classification: city vs. landscape. *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998, pp. 3-8
4. Dorai C. and S. Venkatesh S.: Bridging the semantic gap with computational media aesthetics. *IEEE Multimedia*, 2003, 10, pp. 15-17
5. Cox, J., Miller, M., Minka, T., Yianilos, P. An Optimized Interaction Strategy for Bayesian Relevance Feedback. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998 , pp. 553-558
6. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval. *IEEE Tran. Circuits and Systems for Video Technology*, 1998, Vol. 8, No 5, pp. 644-655
7. Jing, F., Li, M., Zhang, H.-J., Zhang, .B. Relevance Feedback in Region-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, Vol. 14, No. 5
8. Rui, Y., Huan, T. S., Mehrotra . S. Content-based Image Retrieval with Relevance Feedback in MARS. *Proceedings of IEEE Int. Conf. on Image Processing*, 1997, pp. 26-29
9. Koskela, M., Laaksonen, J., Oja, E. Use of image Subsets in Image Retrieval with Self-Organizing Maps. *Proceedings for International Conference on Image and Video Retrieval*, 2004, pp. 508-516
10. Wu, K., Yap, K.H. Fuzzy relevance feedback in content-based image retrieval. *Proc. Int. Conf. Information and Signal Processing and Pacific-Rim Conf. Multimedia*, Singapore, 2003
11. Tian, Q., Wu, Y., Huang, T.S. Incorporate Discriminate Analysis with EM Algorithm in Image Retrieval. *In Proc. IEEE International Conf. on Multimedia and Expo*, 2000
12. Wu, Y., Tian, Q., Huang, T.S. Integrating Unlabeled Images for Image Retrieval Based on Relevance Feedback. *In Proc. of the 15th Int'l Conf. on Pattern Recognition*, Vol.1, 2000, pp.21-24
13. Müller, K.-R., Mika,S., Ratsch, G., Tsuda, K., Schölkopf, B. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 2001, 12(2), pp. 181-201
14. Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, New York, 1999
15. Vapnik, V. An overview of statistical learning theory. *IEEE transactions on Neural Networks* , 5, 1999, pp. 988-1000
16. Lin, C.-F., Wang, S.-D. Fuzzy Support Vector Machines. *IEEE Transactions on Neural Networks*, Volume: 13, 2 , 2002, pp. 464 -471
17. Wu, X., Srihari,R. Incorporating prior knowledge with weighted margin support vector machines. *Proceedings of the international conference on Knowledge discovery and data mining*, 2004, pp. 326 - 333, ISBN:1-58113-888-9
18. Dorado, A., Djordjevic, D., Pedrycz, W., Izquierdo, E. Efficient image selection for concept learning, *to appear in IEE Proceedings Vision, Image & Signal Processing*, 2005

Leveraging Context for Adaptive Multimedia Retrieval: A Matter of Control

Gary Marchionini

School of Information and Library Science, University of North Carolina at Chapel Hill,
100 Manning Hall, Chapel Hill, NC 27599, USA
march@ils.unc.edu

Abstract. Adaptive systems have not been adopted because people prefer stability and predictability for high-level interactions and because such systems have not been robust enough to handle the variety of goals and tasks that people bring to them. Adaptive features at fine grains of activity have been successfully adopted and this paper considers the possibilities of using a variety of contextual features for adaptive multimedia retrieval. It argues that people must gain trust in such systems and be provided with control over the adaptive features within a more general human-computer information retrieval framework.

1 Introduction

Information retrieval (IR) research and development have enjoyed enormous success over the past decades, however, it seems clear that new kinds of thinking and approaches are needed to advance the field to the next levels of theory and practice (Allen et al., 2003). Among the research directions the IR community are pursuing, multimedia retrieval, personalization, context-based retrieval, human information behavior, and user interfaces all intersect under the rubric of human-computer information retrieval (HCIR). HCIR looks at retrieval from the perspective of an active human with information *needs*, information *skills*, powerful digital library *resources* (with dynamic contents that include other humans, sensors, and computational tools), and situated in global and local connected *communities*, all of which *evolve* over time. The goal of HCIR is to integrate human and system interaction so that people are continuously engaged with meaningful information, thus incorporating understanding with finding. HCIR is especially appropriate when the goal is adaptive multimedia retrieval where people and systems are tightly coupled and adaptive.

This paper focuses on HCIR as a socio-technical challenge and argues for placing adaptation control in the hands of humans rather than in automatic systems. In essence, there are two adaptive information retrieval system problems, the technical problem and the human science problem, and I argue that neither is likely to be solved independently. Furthermore, to use a database analogy, the join between these problems is over the common key known as context. More specific to this paper, although multimedia information offers new challenges and opportunities for adaptive IR, the same fundamental limitations of full automatic adaptation apply.

The technical problem involves four main issues: what features of context are useful for retrieval; how can these best be elicited and represented symbolically; how to

combine, weight, fuse or integrate these representations during the retrieval process; and how to evaluate the effectiveness of these contextual features in the overall retrieval process. The human science problem involves three main issues: how to influence human behavior to adopt adaptive systems when there is strong natural inclination toward stable and predictable systems (with the exception of entertainment situations where novelty and suspension of disbelief is valued); how to leverage individual and group characteristics, preferences, and experiences; and how to evaluate the contextual contributions to information seeking progress. To illustrate these problems and argue for human control over adaptation, we consider in turn some elements of context, multimedia IR, and adaptive systems.

2 Context

Only a pure solipsist can ignore context. Whether we think of context broadly as the ground for a primary figure or the environment in which we are inextricably embedded (e.g., Solomon, 1999) or we consider it more narrowly as the most recent queries expressed in an interactive retrieval session, context involves a number of factors, each of which is multi-layered. Seven factors are illustrated here, each of which can be viewed at different levels of granularity. These factors are: human participant and their interactions with other factors over time, goal and the task(s) embedded in it, environment, system, and information corpus. Together, these factors make up the overall capital 'C' Context, with each factor having its own layers of context. From the system side, context is manifested as metadata, from the human science side, it is manifested in the personal and world state. The HCIR challenge is to select the best features from each context factor and leverage those features to better solve the retrieval problem.

The human participant, more specifically called an 'information seeker' and often abbreviated as 'user' is the most problematic contextual factor for multimedia HCIR¹. Characteristics or features of possible interest for multimedia HCIR include physical characteristics such as visual, aural, and kinesthetic abilities and preferences; cognitive characteristics such as knowledge of the problem domain (e.g., whether a search about ionic bonding is by a high school student or an experienced chemist), knowledge about information retrieval (e.g., whether a search is done by a novice or a seasoned reference librarian), and knowledge about the information technology generally and the search system in particular; and affective characteristics such as motivation, mood, and general disposition toward information retrieval as an activity. Each of these user features can be considered at different levels of granularity. For example, a multimedia HCIR system could consider physical characteristics at the neural, muscle group, or whole organism level for the purposes of adaptation depending on whether the feature detection is done through sensor implants, external but obtrusive sensors (e.g., eye tracking, galvanic skin response), or unobtrusive ambient sensors (e.g., video cameras and microphones). Likewise, a system might be tuned to consider overall personality dispositions or instantaneous mood or serotonin levels.

¹ It is important to note that individuals frequently work in groups and thus the group factor should also be included, however, for our purpose here, we limit the discussion to individuals except to the extent that individuals work implicitly with others through retrieval services such as implicit collaborative filtering.

In addition to the personal states, we can leverage personal experiences and interactions over time. Because people are hesitant to literally tune a user profile that defines their personal context, most work has focused on determining such context automatically. The timeframe can be within a single session or across long periods of use. The single session is advantageous in that it presumably is focused on a single topic, however, it is challenging because there are relatively few data points for pattern detection. The long-term timeframe offers more data for pattern detection but may include many goals and tasks with accompanying unique information seeking behaviors. Regardless of the timeframe, these interactions can be used to build a stochastic model of interest and may be modified over time (to account for topic drift). For example, what the information seeker views, annotates, prints, or saves are all indicators of interest that are presumably more important to consider for modeling interest than information that is ignored (Kelly & Belkin, 2004). More global patterns such as application preferences, local system settings, and strategic search patterns may also be used and illustrate the way that different kinds of context might interact and complicate context-based retrieval. Kelly & Teevan (2003) provide a bibliography of work that aims to infer user preferences from their interactions.

Tasks are embedded in goals. Tasks are generally well-defined and map to observable activities. Bates (1983) provides a classic analysis of search tactics and strategies. These activities can be considered at various levels of granularity ranging from keystrokes or mouse moves that map to a specific task like following a hyperlink or typing a query term, to sets of activities that map to strategic tasks such as selecting a system to search or formulating a series of intermediate queries for different topic facets. Automatic spell correction in word processing, automatic term completion in search systems, and automatic URL or email address completions in web browsers and mail applications are well-known examples of techniques that use fine-grained task activity features. Significantly, these techniques have a high probability of being effective and are usually adopted by people. At more general levels of task such as searching for a book or product to buy as part of an e-commerce goal or an alerting service in a research and development setting, recommendations are less highly effective unless the human becomes engaged in actively tuning the service over longer periods of time.

Goals are more global in nature and are the motivations and situations behind searches for information. Goals may be short term (write this paper and find pertinent literature), mid-term (e.g., develop a retrieval system over a three year period), or long-term (e.g., develop my expertise in human-computer information retrieval). They may be related to work or play. They can be urgent or casual. Because goals are often work related and spawn repeated tasks over time, systems can build models for goals over long periods of time if the associated tasks can be isolated from different goal activities. For example, specific goals such as spam filtering are successful because the problem is severe and recurring, and people are willing to do some tuning of the filter model.

Environment includes all the physical and world state conditions. Whether retrieval is being done at home or at work, in private or in public influences behavior and in turn, anything that influences behavior may be useful for leveraging effective retrieval. The world state may also influence retrieval. For example, searching for

'tsunami' before the December 26, 2004 Pacific tsunami yielded quite different results than afterwards. Hence, simply tracking the home pages of major news websites may offer powerful contextual cues to support searches.

Environmental context can be particularly powerful for multimedia retrieval because temporal (e.g., time stamps) and spatial information (e.g., GPS readings), as well as other ambient conditions such as temperature, humidity, and various radiation readings can be automatically captured by the emerging array of media devices. Various sensors can add context beginning with the time of information creation and throughout the information life cycle. Such context is especially helpful in personal collections. For example, Graham et al. (2002) demonstrated the value of time for photography libraries. As such features are combined, even better retrieval precision is expected—it is certainly easy to imagine the positive outcomes of GPS readings in digital photography when added to today's typical time-stamps to photographs. However, the real breakthroughs will come from combining powerful technologies with human effort. For example, face-recognition technology that one takes the time to train for faces of family members can be well-leveraged for personal collections. Systems like PhotoMesa (Kang & Shneiderman, 2000) provide innovative user interfaces that encourage people to invest time in organizing and adding metadata value to image collections.

The system or systems involved in multimedia retrieval are also part of the context at this point in time. If pervasive computing evolves to the point of true cyberinfrastructure that is taken for granted and invisible to users, then systems will become part of the environment, but for the foreseeable future, the systems strongly influence retrieval in both explicit and implicit ways. System features such as computational capability, storage, bandwidth, I/O devices, and firewalls all influence retrieval. Although powerful features may be available on the server side and in the general marketplace, the information seeker may have quite different system capabilities ranging from workstations to cell phones and PDAs. These features are especially important for multimedia retrieval. For example, a music retrieval system that supports 'hum a few bars' queries is useless to an information seeker with no audio input so alternative queries must be supported. Fortunately, determining system features is easily automated as long as users provide permission for environments to be propagated to search services.

The information collection, like the system context can be easily leveraged for retrieval. The intended audience, language, scope and complexity are only a few of the features that may be useful for retrieval. For example, filtering adult content for children is easy to do if these contextual elements are part of the metadata. For multimedia, the media type itself and technical specifications for formats and data creation may support retrieval and presentation. Version or edition also plays a role as content acquires history of use. For example, popularity of retrieval is a feature used for recommendation or retrieval. Thus, hyperlinks and other relationships among information objects and histories of usage are leveraged in various ways for retrieval. More interesting are the annotations and additions made to dynamic media such as blogs and wikis that evolve over time and use. These changes are themselves indicators of interaction (e.g., distributions of annotations across time or space or user community) that should be meaningful for the purposes of retrieval. Active new media that exhibit properties (e.g., display) that are temporally dependent on audience participation

and/or random parameters offer new kinds of retrieval possibilities if those parameters can be captured.

What does such adaptive content mean for retrieval? It makes it both harder and easier. The corpus is not stable---not only is it a changing database but many factors are changing. Typical IR models consider new documents in discrete chunks; however when expensive representation models are used (e.g., LSI), the periods of recomputation may grow long. Adaptive systems imply adaptive information too---the information changes, so not only what might be retrieved tomorrow is a different result set (as if that were not bad enough) but the same information retrieved may no longer behave the same way and thus exhibit a different kind of irrelevance. In the former case, consider a Google query over time that provides different results depending on world events. In this case, the documents have not changed and new ones have been added. However, when we consider adaptive environments such as wikis where not only is new material added but things can actually be removed or edited, then new challenges arise.

3 Multimedia Retrieval

Digital multimedia resources are becoming increasingly common and some argue that they are displacing text information for learning, work, and play. A recent assessment in the US (Oblinger & Oblinger, 2005) reports that 13-17 year olds spend an average of 3.1 hours per day watching TV and 3.5 hours per day with digital media, often doing many kinds of multitasking with the media. Furthermore, more than 2 million children between the ages of 6 and 17 have websites, with girls significantly more likely to have a website. On the research side, Lawrence (2001) offers a provocative argument that technical literature that is not on the WWW is increasingly invisible and irrelevant to new research and development. Ipods and other MP3 players, camera cell phones and digital video and still cameras are a pervasive part of modern culture, which means that more and more multimedia content will be created, managed, and retrieved. In addition to the pervasiveness of forms, the trend toward globalization drives use of non-text content that is perceived and interpreted more viscerally and less symbolically than the written languages of the world.

This trend toward digital multimedia content bodes well for context-based IR because there are more channels and features than might be leveraged for the purposes of retrieval. Context-based IR is especially important to pursue because multimedia user interfaces have not advanced beyond query-by-example styles that do not allow arbitrary queries to be expressed. Furthermore, although there has been enormous investments in content-based retrieval methods for non-text media such as images (e.g., color, texture), spoken language (e.g., speaker alternation patterns), and video (image features, optical flow), evaluations to date consistently favor text-based queries and features for retrieval. For example, the TREC Video Track (Kraau et. al., 2004) has included several variants of features for video retrieval but participants report generally very poor performance with visual features compared to text-based features (e.g., see Christel & Conescu, 2005). Although these results are somewhat disappointing for content-based retrieval, it is important to note that people do like to use visual features and cues. This suggests that we should include

them in retrieval systems as one way to address the human science challenges that mitigate adoption. These limitations of content-based approaches also add to the value of pursuing context-based retrieval for multimedia content. Thus, several forces converge to offer good reasons to apply contextual features to multimedia retrieval.

4 Adaptation

What does adaptive multimedia mean? What adapts and under what conditions? One view is that the system adapts within a session to the conditions in that session; that is, it adapts to a particular person. These adaptations may be stored and analyzed so that subsequent intrasession adaptation can be improved for that person. Another view is that the system adapts over many sessions as it develops a model of the world and various users. In this case the system adapts to the world over time. In either case, adaptation depends on sensing and analyzing the context in which the system works. The discussions of context and multimedia above illustrate many possible sources of evidence that may be used to instantiate adaptive systems. Although there are many technical challenges to solve such as porting profiles across machines and networks (desktop, laptop, PDA/cell phone), the basic human science challenge must also be addressed.

It is reasonable to incorporate fine-grained task-specific adaptations such as automatic query completion because an incorrect completion is easily overridden by simply continuing to type in the query. This is one example of where user control and automatic techniques go hand in hand: there is no additional burden to adopting the automatic action and it is easily or naturally overridden or reversed. Other techniques offer options or alternatives in the periphery. Our Open Video system provides a side panel with popular and related videos in the query results view. This is not in the main user focus and easily ignored. We have chosen to use a static poster frame for these alternative videos rather than fast forwards as that would be too intrusive and distracting. We also provide alternative sort orders for results, but expect users to select the order rather than try to automatically extract a preference from past behavior or similar searches by others (Marchionini & Geisler, 2002). These examples demonstrate giving people control over alternative retrieval parameters and the general trend in design in the post-Microsoft-Bob era.

Adaptation at global system-level scales is more problematic. Although there is a long history of user profiling (e.g., see Brusilovsky and Tasso, 2004 for an introduction and overview), adoption of adaptive systems has always been problematic. People have been hesitant to adopt profiles because of privacy concerns and because we tend to prefer consistency in our work environments unless we fully understand how the changes take place. One challenge is building trust. Without trust, people will not give up control over the adaptation even though control takes time and effort that may diminish efficiency. I suggest that people will adopt adaptive systems that address important or recurring tasks and are well designed, and that they have confidence in, understand, and have optional control over—in short systems that they trust. What are the elements of such trust?

One element of trust is accuracy. The system must work. An adaptive system that provides puzzling or poor results once might be tried again, but a few failures doom it to abandonment. People adopt anti-lock brakes and may perhaps soon accept sleep detection systems because they perform accurately and have important consequences when they work. These automobile systems ‘react’ rather than adapt in that no attempt is made to change future settings of the system to react differently at a later time. Perhaps a better model than adaptation for multimedia information retrieval than adaptive is ‘reactive.’ In a reactive system, dangerous or opportunistic conditions trigger actions in the moment. Users might then be invited to confirm some long-term system adaptation based on the performance of the reaction. A context-based retrieval system that recommends videos I do not want because I happened to order some videos as gifts last week is annoying but forgivable the first time; a retrieval system that leaves out a critical blog entry in a results display because it recalls that I saw that blog entry in a search yesterday is more problematic. The simple solution in the latter case is to clearly provide the previously viewed results as an easily seen and used option. Even more problematic is the system that adjusts what is retrieved (rather than what is displayed) for today’s search based on what was displayed previously. The difficulty is informing the information seeker about how past behavior is being used in the current retrieval situation. People may develop trust in systems where adaptation works well and if they are informed that it is at play and how it operates.

A second element of trust is consistency. We tend to be suspicious of people who behave erratically and there is a delicate balance between boring and wildly creative. I suggest that it is human nature in work-related goals to prefer stable, predictable systems in which we develop trust over time. However, people will accept changes if they improve their work and are easily controlled. People will voluntarily give up control over routine activities if they know that it is their decision to do so and if they trust the system to be capable. People will rebel against even simple control if they believe they are coerced or have no choice. Thus, the human science challenges are about finding a balance between informed consent and the burden of providing it; and in designing easy to use control mechanisms that do not overly burden the information seeker.

5 Conclusion

What does context mean for information retrieval? Context brings more potential evidence, which can be used to help disambiguate queries, customize partitions of the browsing space, populate selective dissemination services that anticipate needs and alert people to potentially pertinent information, and filter out information seen before or that require additional cost (e.g., require translation; require specific fees). On the other hand, context can add more noise to the retrieval and annoy or confuse information seekers. Context also complicates the retrieval process, however given that information retrieval is becoming more sophisticated and our tools more powerful, this complication may be manageable if we are careful and do not promise too much as doing so closes windows of adoption opportunity. An 80% solution may doom an adaptive system unless there is good explicit rationale and user involvement for the other 20%.

Multimedia retrieval can be especially aided by context because there are more features to leverage and the various devices for creating and presenting multimedia typically have computational capabilities built in that can automatically capture many kinds of context. It is vitally important, however, that people be brought more actively into the general retrieval process and that they be consciously involved in context-based retrieval if such techniques are to be adopted. System designers are encouraged to involve people at various stages: assess needs before writing specifications, conduct user studies beginning with prototypes, invite specific feedback over the release life cycle, and especially invite specific feedback over time, and especially inviting people to try context techniques during usage. Most importantly, designers should aim to develop user interfaces and interaction styles that continuously engage information seekers so that they can bring their significant powers of leveraging context to bear on the retrieval problem—that is, they should aim for HCIR rather than IR systems.

Although the net generation may tolerate more multitasking, they will not tolerate more confusion, so fully automatic adaptation is to be discouraged. Adaptive systems without user control are like Proteus—changing in surprising ways that will be confusing and annoying to people. Just as Heracles vanquished Proteus, automatic adaptive systems will be rejected in the marketplace dominated by human science decision making and behavior. Systems that support adaptation through user control will be adopted as people develop trust through exercising this control. Artificial intelligence is a process rather than a goal. It drives ideas about making human endeavor better and more satisfying rather than replacing or emulating it. The mantra for adaptive (or reactive) multimedia information retrieval should be ‘leverage context with abandon, adapt with caution’.

References

1. Allen et al. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, University of Massachusetts Amherst, September 2002. *ACM SIGIR Forum* 37(1), 31 – 47.
2. Bates, M.J. Information Search Tactics. *Journal of the American Society for Information Science*, 30, p. 205-214.
3. Brusilovsky, P. and Tasso, C. (2004) Preface to special issue on user modeling for Web information retrieval. *User Modeling and User Adapted Interaction* 14 (2-3), 147-157
4. Christel, M. & Conescu R. M.. (2005). Addressing the Challenges of Visual Information Access from Digital Image and Video Libraries. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, (Denver, June 2005)*, 69-78.
5. Graham, A., Garcia-Molina, H., Paepcke, A., & Winograd, T. (2002). Time as essence for photo browsing through personal digital libraries. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*. NY: ACM Press. 326-325.
6. H. Kang and B. Shneiderman. Visualization Methods for Personal Photo Collections Browsing and Searching in the PhotoFinder. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2000)*, pp. 1539--1542. IEEE, New York, 2000.
7. Kelly, D. & Belkin, N.J. (2004). Display time as implicit feedback: Understanding task effects. In *Proceedings of the 27th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '04)*, Sheffield, UK, 377-384.

8. Kelly, D. & Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2), 18-28
9. Kraau, W., Smeaton, A., & Over, P. (2004). TRECVID 2004: An Overview. *Proceedings of the Text Retrieval Conference TRECVID Workshop*. Gaithersburg, MD. NIST.
10. Lawrence, S. (2001). Online or invisible? *Nature* 411(6837), p. 521.
11. Marchionini, G. & Geisler, G. (2002). The Open Video digital library. *dLib*, 8(12), <http://www.dlib.org/dlib/december02/marchionini/12marchionini.html>
12. Oblinger, D. & Oblinger, J. (2005). Is it age or IT: First steps toward understanding the Net Generation. In (D. Oblinger and J Oblinger, Eds.) *Educating the net generation*. Educause. <http://www.educause.edu/educatingthenetgen>
13. Solomon, P. (1999). Information Mosaics: Patterns of Action that Structure. In Wilson, T., & Allen, D.K. (Ed.), *Exploring the contexts of information behaviour* (pp. 150-175). UK. London: Taylor Graham.

Rank-Ordering Documents According to Their Relevance in Information Retrieval Using Refinements of Ordered-Weighted Aggregations

Mohand Boughanem, Yannick Loiseau, and Henri Prade

Institut de recherche en informatique de Toulouse,
118 route de Narbonne, 31062 Toulouse cedex 4
{bougha, loiseau, prade}@irit.fr

Abstract. Classical information retrieval methods often lose valuable information when aggregating weights, which may diminish the discriminating power between documents. To cope with this problem, the paper presents an approach for ranking documents in IR, based on a vector-based ordering technique already considered in fuzzy logic for multiple criteria analysis purpose. Moreover, the proposed approach uses a possibilistic framework for evaluating queries to a document collection, which distinguishes between descriptors that are certainly relevant and those which are possibly relevant only. The proposal is evaluated on a benchmark collection that allows us to compare the effectiveness of this approach with a classical one. The proposed method provides an improvement of the precision w.r.t Mercure IR system.

1 Introduction

The purpose of information retrieval (IR) is to find out documents that are relevant with respect to the user's information needs. The most commonly used method is to rank documents according to their relevance to a query stated by the user to represent these needs. The results of the performance evaluation of such a system depends on the rank of relevant documents among those retrieved by the IR system. The method used for rank-ordering the documents is therefore crucial for the result of the evaluation of a query.

In classical information retrieval systems, documents and queries are usually represented by sets of weighted terms. Term weights are computed from statistical analysis. More precisely, the weight of a term in a document is usually estimated by combining the term frequency tf in the document and the inverse document frequency of the term idf [1, 2]. Weights in the query terms, on the other hand, express user preferences.

To evaluate to what extent a document is relevant to a query, a *retrieval status value* (rsv) is computed by aggregating the above weights for the terms present in the query, in a way that reflects the query structure (expressing disjunction or conjunction). Then documents are ranked on the basis of the rsv 's. Different kinds of aggregation functions can be used for combining the weights of the terms (pertaining to the same document) that are present in the considered

query (assumed for the moment to be without any user's preference weighting). Candidate operators for aggregation that are found in the literature are average, similarity-based evaluation, p-norms [3, 4], fuzzy logic conjunctive or disjunctive operations [5, 6, 7]. However, this type of approach leads to a loss of information (e.g. [8]), since individual keyword values are fused together. A consequence is that it is impossible to discriminate documents having the same global relevance value. As an example, let us consider a three-terms query, aggregated by the average. This is only an example, and remarks similar to the ones below apply to other aggregation operators, including *min* and other fuzzy logic connectives. Let us suppose that the evaluation of the query $q = t_1 \wedge t_2 \wedge t_3$ on two documents d_1 and d_2 gives the following results (using normalized weights):

$$\begin{aligned} rsv(q, d_1) &= \frac{w(t_1, d_1) + w(t_2, d_1) + w(t_3, d_1)}{3} \\ &= \frac{0.1 + 0.7 + 0.7}{3} = 0.5; \\ rsv(q, d_2) &= \frac{w(t_1, d_2) + w(t_2, d_2) + w(t_3, d_2)}{3} \\ &= \frac{0.5 + 0.5 + 0.5}{3} = 0.5. \end{aligned}$$

The issue is to know whether the user prefers a document with a medium relevance for all her criteria, or having a high relevance for most of them. This example not only raises an ambiguity problem between documents having apparently the same relevance, but more generally points out the problem of the impact of terms having weights much higher than others. If we want to privilege d_1 over d_2 , this problem can be dealt with by using operators such as Ordered Weighted Average [5], which focus on the weights with high values and model quantifiers such as *most of* [9, 10, 11], provided that such a quantifier is specified in the query. But this does not give a way of preferring d_2 to d_1 if we consider that one low weight can be a serious reason for discounting a document.

In this paper, we try another road. We no longer plan to aggregate the weights, but rather to rank-order the documents directly on the basis of the vectors of the weights of the terms present in the query, using decision making ideas that handle multiple criteria values (here replaced by the relevance value of each query term).

This alternative method is described in section 2. Besides, the possibilistic framework used in the indexation of documents, suggested in [12] (see also [13]), is briefly recalled in section 3. The section 4 presents the results of a large scale evaluation benchmark.

2 Multicriteria Ranking

At least two approaches can be used to compare objects according to multiple criteria. The first one is to aggregate these criteria, then to compare the obtained values. This corresponds to the classical information retrieval approach,

considering each query term relevance as a criterion to fulfil. The second method amounts to compare the criteria evaluation vectors directly by using a refinement of Pareto ordering ($(t_1, \dots, t_n) >_{Pareto} (t'_1, \dots, t'_n)$ iff $\forall i, t_i \geq t'_i$, and $\exists j, t_j > t'_j$). This later method is discussed in this paper. We briefly discuss the aggregation approach first.

2.1 Aggregation Schema

Query terms are usually weighted in order to allow the user to express her preferences and assess the importance of each term. Therefore, the result of the evaluation of a query on a document is a vector of the weights of the terms of the document present in the query, usually modified for taking into account preferences about the importance of the terms in the query. This is why classical IR aggregation methods use weighted conjunction (or disjunction) operators. In conjunctive queries, these operators can be weighted average or weighted minimum. Similar ideas apply to disjunctions as well.

However, this kind of aggregation is too restrictive. To relax the conjunction, ordered weighted operators, such as average (OWA¹ [5]) or minimum (OWmin [14]) have been introduced. The idea underlying this type of aggregation is to give low importance to the smallest weights in the evaluation vector, thus minimizing the impact of small terms, which amounts to model a *most of* quantifier (e.g. [11]).

The *OWmin* operator uses an auxiliary vector of levels of importance in order to minimize the impact of low weighted terms on the final relevance value. Thus, as for OWA, the term weights vectors are ordered and discounted by importance levels, using the minimum instead of the average for computing the global evaluation.

Two weighting methods are considered, based on Dienes implication and on Gödel implication respectively (e.g. [14]). For a vector $T = t_1, \dots, t_n$ representing the indexing weights for a document, t_i is the relevance degree between the i^{th} query term and the document. The vector is assumed to be decreasingly ordered (i.e. $t_i \geq t_{i+1}$). Let $W = (w_1, \dots, w_n)$ be the level of importance vector, also assumed to be decreasingly ordered, i.e. $w_i \geq w_{i+1}$, with $w_1 = 1$. The idea is to give more importance (w_i high) to the terms with a high relevance degree. The *OWmin* aggregation using Dienes implication will be:

$$OWmin_D(T, W) = \min_i(\max(t_i, 1 - w_i))$$

while the Gödel implication is defined by

$$w_i \rightarrow t_i = \begin{cases} 1 & \text{if } w_i \leq t_i \\ t_i & \text{otherwise} \end{cases}$$

which gives:

$$OWmin_G(T, W) = \min_i(w_i \rightarrow t_i)$$

¹ Ordered weighted averaging operators.

In both cases, if the smallest w_i 's are zero, these weighted aggregations amount in practice to restrict the minimum to the t_i 's with high values (since small t_i 's will be replaced by 1 in the aggregation, and the high values of t_i 's, corresponding to values of w_i 's equal or close to 1, will remain unchanged).

However, as already said, we want to rank-order documents by taking advantage of the full weights vector associated with each document, rather than using an aggregated value. This means that we keep the idea of using weights for modifying the indexing weights and restricting the focus of the evaluation, but we no longer compute an aggregated value (taken above as the minimum).

In order to compare vectors, the classical Pareto partial ordering has to be refined, since no pairs of documents should remain incomparable. In the following, we use refinements of the *min* operation, which do refine the Pareto ordering.

2.2 Refining the Minimum Aggregation

Two refinements are considered in this paper, called *discrimin* and *leximin*, see e.g. [15, 16]. They allow to distinguish between vectors having the same minimal value.

Discrimin: Two evaluation vectors are compared using only their distinct components. Thus, identical values having the same place in both vectors are dropped before aggregating the remaining values with a conjunction operator. Thus, only discriminating term weights are considered. In the context of information retrieval, given two vectors representing the weights of terms in query q for documents d_1 and d_2 , expressing term-document relevance. For instance:

$$\begin{aligned} r\vec{sv}(q, d_1) &= (1, 0.5, 0.1, 0.3), \\ r\vec{sv}(q, d_2) &= (0.2, 0.7, 0.1, 1). \end{aligned}$$

Using *min* as an aggregation, these two vectors would get the same score. The *discrimin* procedure “drops” the third term, giving $rsv(q, d_1) = 0.3$ and $rsv(q, d_2) = 0.2$ and allowing to rank these documents.

Leximin: It is a *discrimin* applied on vectors with increasingly re-ordered components. Considering two vectors:

$$\begin{aligned} r\vec{sv}(q, d_1) &= (1, 0.5, 0.1, 0.2), \\ r\vec{sv}(q, d_2) &= (0.2, 0.7, 0.1, 1). \end{aligned}$$

Using the *discrimin*, both values are 0.2. Since the *leximin* sorts the values before comparing them, the 0.2 values are also dropped, giving $rsv(q, d_2) = 0.7$ and $rsv(q, d_1) = 0.5$, thus ranking d_2 before d_1 .

3 Background on Possibilistic Indexing

In this paper, we will use the possibilistic model suggested in [12] and used in [13]. In this approach, the document relevance for the query is no longer given

by the statistical index weight, but by a pair of possibility and necessity degrees computed from it. The retrieval status value is then a pair:

$$rsv(q, d) = (\Pi(q, d), N(q, d))$$

The idea is to distinguish between two aspects of the relevance. If the weight of a term in a document is high enough, then this term is considered to be more or less certainly (or necessarily) representative of the content of the document. If the weight is not sufficiently high, then this term is considered only as possibly representative of the document. Therefore, $rsv(q, d)$ represents to what extent it is possible and certain that d is relevant with respect to q .

To use this possibilistic model, the possibility and necessity degrees of matching between the document and the query terms must be estimated taking into account the statistical weights of the terms in the document. Each document is therefore considered as a fuzzy set of terms (e.g. [17, 9]) by normalizing the weights between $[0, 1]$. A simple, parametrized, way to assess the possibility and the necessity degrees (resp. Π and N) from the *tf*idf* weight w_t of term t in document d once normalized is to use the following piecewise linear transformation:

$$\Pi(t, d) = \begin{cases} 0 & \text{if } w_t = 0 \\ 1 & \text{if } w_t \geq \alpha \\ \frac{w_t}{\alpha} & \text{otherwise} \end{cases} \quad (1)$$

$$N(t, d) = \begin{cases} 1 & \text{if } w_t = 1 \\ \frac{w_t - \alpha}{1 - \alpha} & \text{if } \alpha < 1 \text{ and } w_t \geq \alpha \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Note that when $\alpha = 0$, $\Pi = 1$ and $N = w_t$ and when $\alpha = 1$, $\Pi = w_t$ and $N = 0$.

Thus, the evaluation of a conjunctive query q involving t_1, \dots, t_n amounts to compute the pair of vectors $(\Pi(t_1, d), \dots, \Pi(t_n, d))$ and $(N(t_1, d), \dots, N(t_n, d))$. Then documents are ordered by applying first the leximin/discrimin ranking procedure on the N -vectors, and in case of ties, the leximin/discrimin is applied to the corresponding Π -vectors to try to refine the ordering.

Experiments on the impact of the possibilistic indexing and parameter α on the system performance are now reported.

4 Experimental Results

In this section, we present results of some experiments on a subset of the CLEF2001² collection, to evaluate the merit of the vector-based ranking of documents. Moreover, the impact of the possibilistic encoding of the term weights in the document is first discussed.

4.1 Description of the Experiments

The goal of the experiment is to enhance the global performance of the information retrieval system, and to compare the results that are obtained using

² Cross Language Evaluation Forum: <http://www.clef-campaign.org>

several ranking methods with the ones provided by a classical approach. The first experiment therefore compares the use of the possibilistic framework in the matching process with the classical approach, in order to determine the best α value for the possibilistic encoding of the term weights. The second experiment compares results obtained with several conjunction aggregation operators, namely the weighted sum aggregation underlying the classical approach (used in Mercure [18]), the two defined *OWmin* and the classical minimum, and with the refined leximin/discrimin-based ranking method.

The mercure information retrieval system. To index the collection, and to compare our results with a classical approach-based system, we used Mercure [18]. In this system, the weight w_t of a term for a document is computed using a formula derived from the *Okapi* system [19]:

$$w_t = \frac{tf}{0.2 + 0.7 \times \frac{dl}{\Delta_l} + tf} \times \left(\log\left(\frac{n_{tot}}{n}\right)\right) \quad (3)$$

where tf is the term frequency in the document, dl is the document length, Δ_l is the average document length in the collection, n_{tot} is the size of the collection and n is the number of documents containing the term. In the collection used, we have $\Delta_l = 184.6$ and $n_{tot} = 113005$.

The final similarity degree S_{qd} between a query q and a document d , giving the relevance of the document for the query, is computed as:

$$S_{qd} = \sum_{t \in q} \lambda_t \times w_{td}$$

where λ_t is an importance weight for the term in the query (here always 1) and w_{td} is the index term weight for document d , given by equation 3.

4.2 CLEF Collection

The collection used in this experimentation is the English part of the CLEF2001 collection, containing 113,005 articles from the 1994 *Los Angeles Times*.

During the indexing stage, terms frequencies are computed for each document. These terms are stemmed using the Porter algorithm [20], and stop-words (i.e. words that bring no information) are removed.

Together with the collection of documents, a set of topics, which are evaluated on the given documents by human experts, are available. These topics, identified by a number, are described by a title, a short description of the topic, and a narrative part giving precise relevance criteria. They are used as a basis for generating the queries to be evaluated by the IR system. Moreover, the documents estimated to be relevant by experts are provided for each topic.

As an example, the topic 41 is defined as:

title: Pesticides in Baby Food

description: Find reports on pesticides in baby food.

narrative part: Relevant documents give information on the discovery of pesticides in baby food. They report on different brands, supermarkets, and companies selling baby food which contains pesticides. They also discuss measures against the contamination of baby food by pesticides.

4.3 Evaluations and Results

To evaluate the system, we used a set of 25 queries automatically built from the descriptive part of the CLEF topics, considered as keywords conjunctions.

To estimate the quality of the information retrieval system, two measures are used. The recall is the ratio of relevant documents retrieved to those relevant in the collection, and the precision is the ratio of relevant documents among the documents retrieved. Since the precision at x , denoted Px , which is the ratio of relevant documents in the x first retrieved documents, is easier to estimate, it is usually used to represent the system performance. Precisions at 5, 10, etc. noted P5, P10, are thus computed. The average precision (AvgPr) is the average of the relevant documents precisions. The precision of a document is the value of Px , for x taken as the rank of the considered document in the retrieve list. The given values are averages on the evaluated queries results.

Possibilistic degrees. The first experiment uses the possibility and necessity degrees to estimate the relevance of the documents for the queries, to compare this approach with the classical one of the Mercure system. The aggregation of individual query terms is done using the *min* operator. Results of the possibilistic approach are shown in figure 1(a), and those of the Mercure system are reported in table 1.

Table 1. Precision of the Mercure system

P5	P10	AvgPr
0.3909	0.3682	0.3827

First of all, it can be noticed that we obtain the same precision value for $\alpha = 0$ and $\alpha = 1$, since this is equivalent as $\Pi = 1, N = w_t$ and $\Pi = w_t, N = 0$ respectively. The final ranking is therefore done using only the w_t 's. However, in this case, the precision is lower than for the classical system. This is not surprising since the aggregation is done using the minimum (without leximin refinement) in place of the sum. This aggregation is indeed too coarse, justifying the use of refined methods, such as *OWmin* that we will present in the following.

However, whereas the average precisions are lower than what is obtained by the classical method, it is worth noticing that P5 is higher for $\alpha = 0.2$ and 0.3 . Figure 1(b) shows the number of terms in the index for each value of w_t . Most of the terms have a weight around 0.2. Indeed, the conversion to possibility and necessity degrees has a stronger impact for values of α that share the terms ratio distribution in two approximately equal parts. Then more terms may be discriminated. As there are very few terms with w_t higher than 0.5, for such values of α the terms are discriminated only by Π or only by N , and the precision tends to be the one corresponding to $\alpha = 1$.

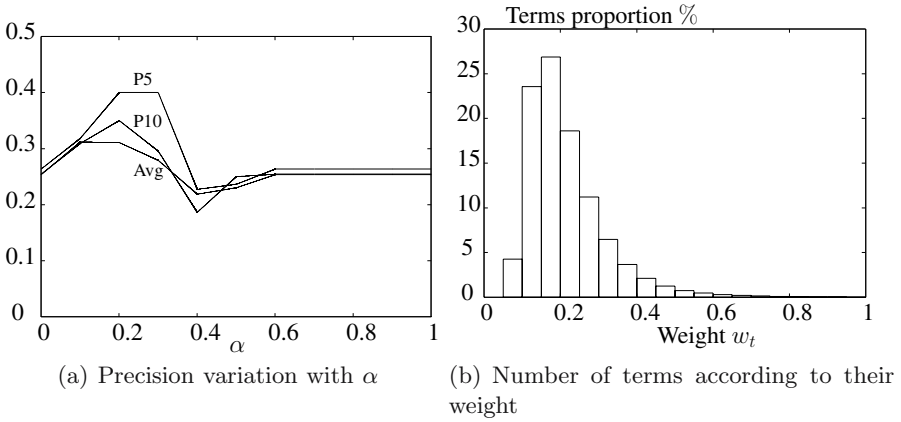


Fig. 1. Interpretation of α values

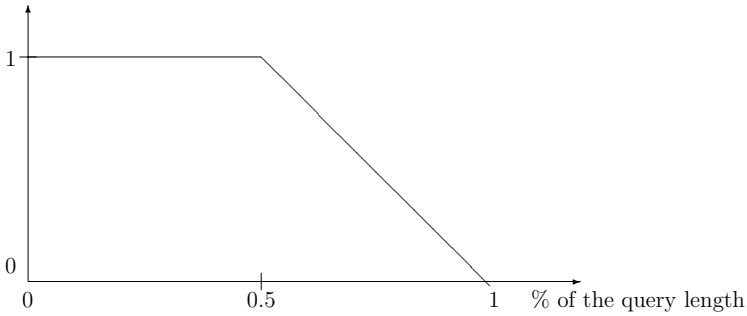


Fig. 2. Model of the *most of* operator

Thus, using a proper tuning of α and the coarse *min* aggregation, the method based on possibilistic degrees is still comparable to the classical approach based on statistical weights and weight sum. However, this improvement depends on the value of α , whose the optimal value depends on the terms repartition in the collection. Therefore, this value of α is collection dependent. Some other experiments should be done using other collections in order to estimate the collection effect on the performances.

Comparison of ranking methods. We will now evaluate the ranking method presented in sections 2 and 3 w.r.t. the one used in Mercure. To apply these ordered weightings, the vectors containing the weights of each query term in the document are decreasingly ordered. As the queries considered here do not introduce further preference levels, the ordered vectors are then weighted using a kind of *most of*-like operator as in section 2.1, based on Dienes or Gödel implications. This type of operator gives more importance to the highest term weights, minimizing the impact of the lowest ones. The weighting vector is computed according to the query length, using the function pictured in figure 2. Results are then sorted using this (II, N) values modified by the weight w_i as in 2.1.

Table 2. Comparison of multicriteria methods

α	Ranking method	Rounding P5 decimal	P10	AvgPr
0.1	leximin + OW_D	1	0.4273 0.3682	0.3572
0.1	leximin + OW_G	1	0.4273 0.3636	0.3571
0.6	leximin + OW_D	5	0.4273 0.3500	0.3161
0.6	leximin + OW_D	4	0.4273 0.3500	0.3158
0.6	leximin + OW_D	6	0.4273 0.3500	0.3152
0.8	leximin + OW_D	2	0.4273 0.3409	0.3209
0.1	leximin	1	0.4182 0.3545	0.3532
0.2	leximin + OW_G	2	0.4182 0.3545	0.3132
0.2	leximin + OW_D	2	0.4000 0.3409	0.3216
Mercure sum			0.3909 0.3682	0.3827

When two documents have the same relevance value, we compared them using the leximin ordering method.

Moreover, the numerical precision of term degrees is not meaningful, since resulting from the normalization and the possibilistic transformation, which leads some values to differ only at the fifth decimal. The possibilistic degrees used between terms and documents have therefore been rounded. The discrimin/leximin results depending on this rounding, several precision levels have been tested to estimate the impact of the rounding on the system performances.

As in the previous evaluations, we used 25 queries, using different α values, rounding precision (i.e. the number of decimals kept), aggregating and ranking methods, to estimate the document relevance degrees.

Table 2 shows the best P5 results, compared with the classical approach using statistical weights and the sum. Results using discrimin do not appear since they are not as good as those obtained with leximin.

It should be noted that here, the better results are obtained for values of α different from the previous one. Therefore, the optimal value for this parameter depends not only on the term distribution in the collection, but also on the methods used to aggregate and sort the results. As expected, there is almost no performance difference between the two weighting techniques of section 2.1, denoted OW_G and OW_D in table 2.

Results are rather promising, since they are already better than the ones obtained with the standard ordering method on possibilistic degrees, based on *min*. Moreover, the best results are even better than those of the classical approach, which was our baseline here, improving P5 up to 9.3%. Nevertheless, the average precision is lower. As there is only few relevant documents in the collection for each query (about ten), the ordering method loses its effect for P_n with n rising, since this value is estimated by counting the relevant documents in the n firsts, whatever their position. Thus, the retrieved relevant documents are in the top of the list. The ranking method has a strong effect on the system performances. Moreover, the presented results are averages on the results obtained for 25 queries. The fact that P5 is better than with the classical approach whereas P10 is lower means that some queries are improved while other are degraded,

and that the improvement is higher than the degradation. This system is therefore suitable for high precision evaluation, where good precision combined with low recall is desirable. Indeed, it improves the number of relevant documents retrieved in the top of the list, but can miss some relevant documents at a lower rank. This improvement is thus obtained to the detriment of the average performances. This entails that other performance measures that handle non-binary relevance assessment, such as cumulative gain [21], would be more appropriate than precision and recall. The use of precision and recall was motivated by the need to compare performance with other systems, and the fact that the CLEF collection only supplies binary relevance results. Moreover, in a realistic information retrieval system, such as web search engine, only the first retrieved documents are of interest for the user, as she rarely browse through more than 10 results (which is often the default number of results by page displayed).

5 Conclusion

In this paper, we have presented a new approach to rank documents according to their relevance, using flexible aggregation methods and refined vector-based rank ordering methods. This approach was evaluated on a subset of the CLEF2001 collection. We compared the refined rank-ordering approach (possibly using some ordered weighting method) with the classical approach based on relevance scores aggregated by a weighted sum. These experiments suggest the effectiveness of the refined rank-ordering approach, as it outperforms sum or min-based aggregation methods to some extent.

These first preliminary results indicate that ranking documents can take advantage of the full weights vector, rather than using an aggregated value. In future works, we plan to evaluate the approach on larger collections, such as TREC collections, and secondly to explore other variants of the flexible aggregation/ranking techniques. Indeed, the statistical result of system performance are heavily dependent on the collection. Moreover, the techniques explored, from the decision making field, are only a subset of the one available.

This approach is not restricted to textual IR, but could be applied to any documents retrieval system using several criteria for describing them, such as in picture or audio sources.

In this approach, the leximin refinement of the minimum aggregation has been used to rank-order documents. It could be possible to use a counterpart of this idea applied to the sum. It amounts to the lexicographic ordering of vectors of the form $(t_1, t_1 + t_2, \dots, t_1 + t_2 + \dots + t_n)$ for $t_1 \geq t_2 \geq \dots \geq t_n$. This ordering is known as Lorenz dominance (see, e.g. [22]).

References

1. Grossman, D., Frieder, O.: Information Retrieval: Algorithms and Heuristics. Kluwer Academic Publishers (1998)
2. Salton, G., McGill, M.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)

3. Salton, G., Fox, E., Wu, H.: Extended boolean information retrieval. *Communications of the ACM* **26** (1983) 1022–1036
4. Robertson, S.E.: The probability ranking principle in IR. *Journal of Documentation* **33** (1977) 294–304
5. Yager, R.: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics* **18** (1988) 183–190
6. Dubois, D., Prade, H.: A review of fuzzy sets aggregation connectives. *Information Sciences* **3** (1985) 85–121
7. Fodor, J., Yager, R., Rybalov, A.: Structure of uni-norms. *International Journal of Uncertainty, Fuzzyness and Knowledge Based Systems* **5** (1997) 411–427
8. Schamber, L.: Relevance and information behavior. In: *Annual Review of Information Science and Technology*. Volume 29. Medford, Learned Information, INC. (1994) 3–48
9. Kraft, D., Bordogna, G., Pasi, G.: Fuzzy set techniques in information retrieval. In: *Fuzzy Sets in Approximate Reasoning and Information Systems*. Kluwer Academic Publishers (1999) 469–510
10. Bordogna, G., Pasi, G.: Linguistic aggregation operators in fuzzy information retrieval. *Int. J. Intell. Syst.* **10** (1995) 233–248
11. Losada, D., Díaz-Hermida, F., Bugarín, A., Barro, S.: Experiments on using fuzzy quantified sentences in adhoc retrieval. In: *Proc. of The 2004 ACM Symp. on Applied Computing, SAC 2004, Nicosia, Cyprus*, ACM Press N.Y. (2004) 1059–1064
12. Prade, H., Testemale, C.: Application of possibility and necessity measures to documentary information retrieval. *LNCS* **286** (1987) 265–275
13. Boughanem, M., Loiseau, Y., Prade, H.: Graded pattern matching in a multilingual context. In: *Proc. 7th Meeting Euro Working Group on Fuzzy Sets, Eurofuse, Varena* (2002) 121–126
14. Dubois, D., Prade, H.: Semantic of quotient operators in fuzzy relational databases. *Fuzzy Sets and Systems* **78** (1996) 89–93
15. Dubois, D., Fargier, H., Prade, H.: Beyond min aggregation in multicriteria decision: (ordered) weighted min, disci-min, leximin. In Yager, R., Kacprzyk, J., eds.: *The Ordered Weighted Averaging Operators*. Kluwer (1997) 181–192
16. Moulin, H.: *Axioms of Cooperative Decision-Making*. Cambridge University Press (1988)
17. Buell, D.: An analysis of some fuzzy subset applications to information retrieval systems. *Fuzzy Sets and Systems* **7** (1982) 35–42
18. Boughanem, M., Dkaki, T., Mothe, J., Soule-Dupuy, C.: Mercure at TREC-7. In: *Proc. of TREC-7*. (1998) 135–141
19. Robertson, S.E., Walker, S.: Okapi-keenbow at TREC-8. In: *Proc. 8th Text Retrieval Conf., TREC-8* (1999) 60–67
20. Porter, M.: An algorithm for suffix stripping. *Program* **14** (1980) 130–137
21. Järvelin, K., Kekäläinen, J.: Ir evaluation methods for retrieving highly relevant documents. In Belkin, N., Ingwersen, P., Leong, M.K., eds.: *Proc. of the 23rd ACM Sigir Conf. on Research and Development of Information Retrieval*, Athens, Greece, ACM Press, N.Y (2000) 41–48
22. Dubois, D., Prade, H.: On different ways of ordering conjoint evaluations. In: *Proc. of the 25th Linz seminar on Fuzzy Set Theory*, Linz, Austria (2004) 42–46

Ranking Invariance Based on Similarity Measures in Document Retrieval

Jean-Francois Omhover, Maria Rifqi, and Marcin Detyniecki

LIP6 – Pole IA, 8, rue du Capitaine Scott, 75015 Paris, France

Abstract. To automatically retrieve documents or images from a database, retrieval systems use similarity measures to compare a request based on features extracted from the documents. As a result, documents are ordered in a list by decreasing correspondance to the request. Several comparison measures are used in the field and it is difficult to choose one or another. In this paper, we show that they can be grouped into classes of equivalent behavior. Then, in a query by example process, the choice of these measure can be reduced to the choice of a family of them.

Keywords: Fuzzy Similarity Measures, Image Retrieval, Aggregation, Segmentation.

1 Introduction

Decriptions, queries and similarity measures are the three basic components of a document retrieval system. In the case of an image retrieval system, an image is indexed through a visual description (color, shape, texture,...) by means of a vector or a set of features. The query part of a retrieval system consists, for the user, in choosing an image or a part of image example. Then the image retrieval system evaluates the similarity between this request and each image of the database (or a part of it). This is done by computing a similarity measure between pairs of descriptions. The comparison of two image descriptions is therefore a fundamental operation for such systems. Obviously then, the choice of a particular similarity measure is a crucial point. In response to its request, the user gets a list of images. This list is ordered by the decreasing degree of similarity to the request.

When a user obtains this list, he tends to ignore the similarity degrees of the images and focuses on the images themselves. What is important to him is the order of this list of documents [10, 8], because he will evaluate the relevance of each resulting images in their order of arrival. The similarity degree is not examined by the user. Sometimes, it is not even displayed. This similarity degree is also discarded in the measures evaluating the efficiency of information retrieval systems. In particular, the recall and precision measures are based on the number of relevant documents in the first results of the system. Hence, they depend on the order in which these relevant documents appear, and not on the relevance values computed by the system for these documents.

As a matter of fact, the value of the similarity itself is unimportant for both the user and the system. The system only uses the value to order the results. The user barely notices this value because his attention is focused on the content of the result images. Based on this fact, choosing between one measure or another to compare visual descriptions is of little interest if two measures lead to the same ordered list.

Very different kind of similarity measures are used in the field. In our image retrieval system [6], many similarity measures can be used to support various queries. In particular, we use the fuzzy similarity measures to compute the similarity between gradual visual descriptors as those used classically in the field. This family of measures, first introduced in a psychological context [14], were adapted to gradual sets [2]. Fuzzy similarity measures can cover various intuitive user needs. They can also be adapted to various descriptors. As an example of the measures falling under this formalism, the classical histogram intersection [12] has widely spread in the image retrieval community.

In this paper we study the set of similarity measures in the perspective of ordering documents relatively to a request. We show that, if the order of the results of a system is indeed the only information to be considered for its use and evaluation, then similarity measures can be grouped in classes of equivalent behavior. For any given request, measures belonging to a same group do provide the exact same result order. We also show that the value of one measure can be predicted from the value of one of its equivalent measures so that we can predict the outcome of a procedure based on similarity values rather than results order. We also study the consequences of this result in the context of image retrieval.

We first introduce the similarity measures in a CBIR perspective (see section 2). We then build a formal theory about order invariance for fuzzy similarity measures (see section 3). The three definitions introduced in this section let us draw equivalence classes that group the similarity measures leading to the same orders. Then, as an application and an illustration, we focus on a specific set of similarity measures, that are Tversky's ratio model measures (see section 4). In this set of measures, we entirely describe the families of equivalent measures. In the last section (section 5) we discuss the consequences of considering equivalent measures on the issue of document retrieval by similarity.

2 Similarity Measures for Image Retrieval Systems

Any image retrieval system bases its action on the computation of similarity measures. Commonly, histogram intersection measures [12], or distance models are widely used in this field [3, 5, 11]. Generalising and covering different similarity measures (as histogram intersection), we use fuzzy similitude measures as a tool to compare image indexes and to evaluate visual similarities.

We first introduce the definition of fuzzy similitude measures, then we apply some of these measures to the comparison of global histograms. Finally, we observe an invariance in the ranking provided by different fuzzy similarity measures. It is not the purpose of this paper to focus on the image representation we use to compare the images.

2.1 Tversky’s Ratio Model

Measures of similarity (or dissimilarity) are distinguished into two classes: the geometric one and the set-theoretic one.

Geometric distance models are the most commonly used approach. Objects to be compared are considered as points in a metric space. These models are constrained by 4 properties that a distance has to satisfy: positivity, symmetry, minimality and triangular inequality.

These axioms were in particular studied by [14], who proposed an approach based on more psychological considerations. His study concludes on the very questionable character of the distance axioms. Other studies pointed out the problematic behavior of distances in high dimensional feature spaces [4, 1]. Tversky proposed a set-theoretic definition of similarity. In his scheme, objects to be compared are described by means of sets of binary features. Let a and b be two objects described respectively by the sets of features A and B , and s a measure of similarity, then $s(a, b) = F(A \cap B, A - B, B - A)$ with F a real function of three arguments: the common features ($f(A \cap B)$) and the distinctive features ($f(B - A), f(A - B)$).

His mathematical formulation (called the Ratio Model) introduces a ratio between common features and distinctive features. The two parameters α and β are real positive weights that balance the influence of each part of the distinctive features (those shared by A and not by B and those shared by B and not by A):

$$S(a, b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(B - A) + \beta f(A - B)}$$

with $\alpha, \beta \geq 0$, and f an additive interval scale.

2.2 Similitude Measures for Fuzzy Sets

Because of the restriction of the contrast model to binary features, we proposed in [2, 9], a generalisation of Tversky’s model to fuzzy features¹. Furthermore, our framework enables to study particular families of similarity measures according to additional properties corresponding to particular needs and behaviours.

In this framework, for any set Ω of elements, $P_f(\Omega)$ denotes the set of fuzzy subsets of Ω and a fuzzy set measure M is supposed to be given such that $M : P_f(\Omega) \rightarrow \mathbb{R}^+$ and $M(\emptyset) = 0$ and M is monotonous with respect to \subseteq (for instance $M(A) = \sum_{count} f_A(x)$).

Definition 1. *An M -measure of comparison S on Ω is a mapping $S : P_f(\Omega) \times P_f(\Omega) \rightarrow [0, 1]$ such that*

$$S(A, B) = F_S(M(A \cap B), M(B - A), M(A - B))$$

where F_S is a mapping $F_S : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [0, 1]$ and M a fuzzy set measure on $P_f(\Omega)$.

We denote $X = M(A \cap B), Y = M(B - A), Z = M(A - B)$.

¹ Latter attempts of generalisation to fuzzy features can be found after the publication of our work: see [10], [13].

A measure of comparison captures various families of measures. We are interested in those which evaluate the likeliness of two descriptions. We have called them *measures of similitude*.

Definition 2. An M -measure of similitude S on Ω is an M -measure of comparison S such that $F_S(X, Y, Z)$ is :

- monotonous non decreasing with respect to X
- monotonous non increasing with respect to Y and Z .

2.3 Image Retrieval by Fuzzy Ressemblance

Our image retrieval system [7, 6] is based on a regional representation of the images. Regions are extracted automatically by a segmentation algorithm. For each image of an image database, a description of each of its region is computed and stored. The image retrieval can then be driven on the basis of regional queries. The comparison between the descriptors of these regions is computed using an M -measures of similitude. These measures let us build intuitively modifiable queries, and can be aggregated homogeneously to form composite requests (requests for images containing several regions of interest) [6].

For the sake of simplicity, we focus here on the classical global representation (i.e. descriptions of a non segmented image). We represent each image by its global histogram [12], which gives the distribution of pixels' colors for a given color palette C_1, \dots, C_n . It can be considered as a fuzzy set membership function H_I on the universe C_1, \dots, C_n . To compute similarities between images, we used fuzzy M -measures of ressemblance to compare their histograms. For two images I_1, I_2 , the comparison is done by computing one of the four following measures based on their histograms H_{I_1}, H_{I_2} , we denote $X = M(H_{I_1} \cap H_{I_2})$, $Y = M(H_{I_2} - H_{I_1})$, $Z = M(H_{I_1} - H_{I_2})$, M being the area of the given set:

$$\begin{aligned}
 S_{jaccard}(X, Y, Z) &= \frac{X}{X+Y+Z} \\
 S_{dice}(X, Y, Z) &= \frac{2X}{2X+Y+Z} \\
 S_{ochiai}(X, Y, Z) &= \frac{X}{\sqrt{X+Y}\sqrt{X+Z}} \quad \text{with } F_{FD}(\phi) = \frac{1}{1+\exp\left(\frac{\phi-\phi_0}{\Gamma}\right)} \text{ and} \\
 S_{Fermi-Dirac}(X, Y, Z) &= \frac{F_{FD}(\phi) - F_{FD}\left(\frac{\pi}{2}\right)}{F_{FD}(0) - F_{FD}\left(\frac{\pi}{2}\right)}
 \end{aligned}$$

$\phi = \arctan\left(\frac{Y+Z}{X}\right)$, Γ is a positive real and $\phi_0 \in [0, \frac{\pi}{2}]$. The parameter ϕ_0 controls the point where the decrease will occur whereas Γ controls the decrease speed of the measure. These measures generalize the classical similarity measures to fuzzy sets, well-known in information retrieval particularly.

2.4 Order Induced by a Similarity Measure

In the following sections, we will study the invariance of the order observed within images ranked by their similarity to a query. Such an invariance can be observed on the two simple requests shown on the figure 1. Images are described by a simple global histogram, then two different image requests are given. To

answer these two different requests we evaluate In the following sections, we will study the invariance of the order observed within images ranked the similarity of each entry of our image database (Washington Groundtruth) by means of two different similarity measures, that are Jaccard and Dice measures presented in section 2.3. For each query on the figure 1, we clearly see that the results obtained by the two different similarity measures are ordered the same way. Practically, for one user these two results are the same.


Formally, for a database of images I_1, \dots, I_n , the value returned by a measure S for the comparison of each entry to a query R is used to order I_1, \dots, I_n by decreasing resemblance to R .

A similarity measure as defined in 2.2 is computed based on three real values X, Y and Z (as denoted in definition 1). Then the problem of ordering pairs of images by their similarity is extended to the problem of ordering real triplets of values (X, Y, Z) by a measure of similarity S .



Request 1 with Jaccard:

Image11.jpg (1/80) method0 = 1.000000(1)	Image12.jpg (2/80) method0 = 0.794894(2)	Image26.jpg (3/80) method0 = 0.500621(3)	Image25.jpg (4/80) method0 = 0.500463(4)	Image13.jpg (5/80) method0 = 0.498674(5)
				

Request 1 with Dice:

Image11.jpg (1/80) method0 = 1.000000(1)	Image12.jpg (2/80) method0 = 0.885728(2)	Image26.jpg (3/80) method0 = 0.667219(3)	Image25.jpg (4/80) method0 = 0.667078(4)	Image13.jpg (5/80) method0 = 0.663487(5)
				

Request 2 with Jaccard:

Image10.jpg (1/80) method0 = 1.000000(1)	Image09.jpg (2/80) method0 = 0.678558(2)	Image05.jpg (3/80) method0 = 0.629575(3)	Image08.jpg (4/80) method0 = 0.629084(4)	Image27.jpg (5/80) method0 = 0.556125(5)
				

Request 2 with Dice:

Image10.jpg (1/80) method0 = 1.000000(1)	Image09.jpg (2/80) method0 = 0.808501(2)	Image05.jpg (3/80) method0 = 0.772686(3)	Image08.jpg (4/80) method0 = 0.772316(4)	Image27.jpg (5/80) method0 = 0.714756(5)
				

Fig. 1. Two 5-best-results lists using Jaccard and Dice measures applied to image histograms

This problem is not only relevant for the simple purpose of global histogram comparison, but also for any fuzzy set representation, or multiple sets of features. For example, in our CBIR system, similarity measures are computed to match regions in each image I_i with regions in a given query R . As we use a best-matching mechanism, the order induced by the similarity measure between pairs of regions in I_i and R significantly influences the result of the matching.

3 Classes of Equivalent Similarity Measures

Three formal definitions can be proposed for the equivalence relations between similarity measures based on order conservation. These are the two first :

Definition 3. For any similarity measures S_a and S_b , S_a is "equivalent in order" to S_b if and only if

$$\begin{aligned} &\forall (X, Y, Z) \in \mathbb{R}^{+3}, \forall (X', Y', Z') \in \mathbb{R}^{+3} \\ &S_a(X, Y, Z) \leq S_a(X', Y', Z') \\ &\iff S_b(X, Y, Z) \leq S_b(X', Y', Z') \end{aligned}$$

Definition 4. For any similarity measures S_a and S_b , S_a is "equivalent by a function" to S_b if and only if there exists a strictly increasing function:

$$f : \begin{cases} Im(S_a) \rightarrow Im(S_b) \\ x \mapsto f(x) \end{cases}$$

such as $S_b = f \circ S_a$
and $Im(S_t) = \{\alpha / \exists (X, Y, Z) \in \mathbb{R}^{+3}, \alpha = S_t(X, Y, Z)\}$.

These two relations are obviously reflexive, symmetrical and transitive. They define equivalence classes within similarity measures. It is also a well known fact that these two definitions are equivalent, meaning that two measures are equivalent in order if and only if they are equivalent by a function.

In a CBIR perspective, what is interesting here is that two measures S_a and S_b that obtain the same results order are linked one to the other by a bijective

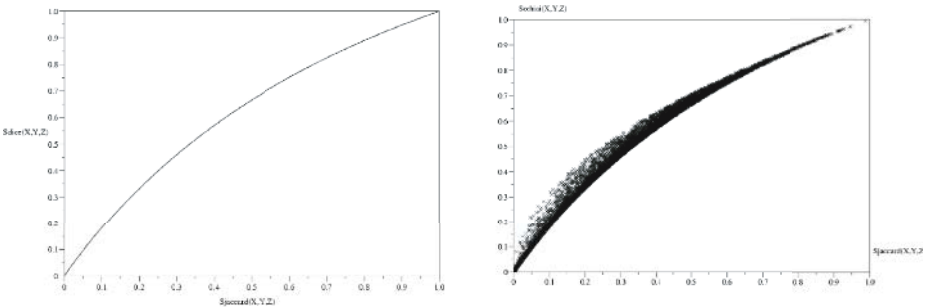


Fig. 2. (a) $S_{jaccard}(X, Y, Z)$ vs $S_{dice}(X, Y, Z)$ (b) $S_{jaccard}(X, Y, Z)$ vs $S_{ochiai}(X, Y, Z)$

function. This means that the value computed by S_b for the evaluation of the similarity of two documents can be predicted from the value computed by S_a for the same purpose. This observation puts this reflexion beyond our preliminary hypothesis : if, ever, the order is *not* the only important information in a result list (as if the results are thresholded by their similarity value, for example), we can still predict the values taken by a measure from any of its equivalent measures. We can then predict the outcome of a procedure relying on the value of the similarity measure (see section 5).

Thanks to these first two definitions of equivalence, we can show that Jaccard, Dice and Fermi-Dirac measures belong to the same equivalence class and that Ochiai's measure belongs to a different class. The figure 2 illustrates the fact that $S_{jaccard}$ can be written as a function of S_{dice} and can not be written as a function of S_{ochiai} : a single value of $S_{jaccard}$ corresponds to many values of S_{ochiai} (and vice versa).

A "value versus value" plot such as figure 2 offers a convenient representation of two measures equivalence. By this kind of graphic, we can simply detect the equivalence of two similarity measures.

The third definition is based on the level sets of the similarity measures. We denote the level set of a similarity measure S_i at the level λ by S_i^λ :

$$S_i^\lambda = \{(X, Y, Z) / S_i(X, Y, Z) = \lambda\}$$

Definition 5. For any similarity measures S_a and S_b , S_a is said "equivalent in level-sets" to S_b if:

$$\forall \beta \in Im(S_b), \exists! \alpha \in Im(S_a) \text{ such that } S_a^\alpha = S_b^\beta$$

This definition means that measures that are equivalent in level sets have a common structure of level sets : they rely on the same level sets but maybe on different levels. Here again we rely on the fact that the value of the similarity is unimportant. This relation is obviously reflexive. It can easily be proven that it is also symmetrical and transitive. We have shown that in the case of continuous measures (as are most of the similarity measures used in the field), this third definition was equivalent to the two first definitions. This can be shown thanks to the monotonicity property of the similarity measures for each of their variables (X, Y, Z) .

As a result, we have three definitions leading to the same notion of equivalence. It means that measures that induce the same order in every result list are related one to the other by a function, and their level sets have identical shapes.

The notion of equivalence between similarity measures leads to the construction of equivalence classes. Each class gathers similarity measures that are equivalent one to the other. As a result, families of similarity measures can be drawn ; these families correspond to the sets of similarity measures that induce the same order within the results of a query-by-example process.

4 Application to the Equivalence of Tversky's Similarity Measures

As an application of our theory, we propose to study the form of equivalence classes within a particular set of measures: the similarity measures of Tversky's ratio model. This model proposed in [14] gathers different behaviors by means of two weights balancing the influence of the two sets of distinctive features in the similarity measure. In this section, we give, for this family of measures, a complete characterisation of the equivalence classes defined in the previous section.

4.1 Tversky's Ratio Model Measures and Their Behaviours

As introduced in section 2.1, Tversky proposed a general expression for the computation of the similarity, the ratio model:

$$S_{(\alpha,\beta)}(X, Y, Z) = \frac{X}{X + \alpha Y + \beta Z}$$

This formulation gives two free parameters (α, β) . The choice of a given couple of parameters (α, β) leads to a particular measure behaviour, for example:

- if $\alpha = \beta$, the measure is symmetrical. Actually, it is a ressemblance measure (see [2]). Jaccard's measure (histogram intersection) and Dice's measure are two examples of Tversky's ratio model measures ($S_{(1,1)} = S_{jaccard}$, $S_{(\frac{1}{2}, \frac{1}{2})} = S_{dice}$).
- for any α , if $\beta = 0$, the measure is called an inclusion measure (see [2]), and it evaluates the degree of inclusion of A in B .
- for any β , if $\alpha = 0$, the measure is called a satisfiability measure (see [2]), and it evaluates the inclusion of B in A . This kind of measure is used for instance, in decision analysis to evaluate the satisfiability of the observation of a fact B for the premise A of a rule.

4.2 Balancing Parameter of Tversky's Ratio Measures

As we have shown in section 3, the equivalence of two measures can be determined by a study of their level sets. Let us consider two measures $S_{(\alpha,\beta)}$, $S_{(\alpha',\beta')}$, and two levels h, h' .

To prove the equivalence of $S_{(\alpha,\beta)}$ and $S_{(\alpha',\beta')}$, we have to show that, for any h' , we can find an unique h such as $S_{\alpha,\beta}^h = S_{\alpha',\beta'}^{h'}$. We can show that it happens if and only if $\alpha.\beta' - \alpha'.\beta = 0$.

So, two Tversky's measures $S_{(\alpha,\beta)}$ and $S_{(\alpha',\beta')}$ are equivalent if and only if $\alpha.\beta' = \alpha'.\beta$. In other words, two Tversky's Ratio measures are equivalent if their parameters have the same ratio $\frac{\alpha}{\beta}$.

If we are interested only in the order induced by a similarity measure taken within Tversky's ratio model, then the choice of the parameters α and β can be reduced to the choice of a single parameter $k = \frac{\alpha}{\beta}$:

- $k = 0$ for an inclusion measures.
- $k = 1$ for a ressemblance measure.
- $k = +\text{inf}$ for a satisfiability measure.

5 Discussion

This section studies the consequences of using measures taken in a same given equivalence class, in other words, the implications of order invariance in some applications.

5.1 Order Based Procedures Such as Recall and Precision

The first consequence lies in the document retrieval context, where resemblance measures are used to compare features extracted from the documents. As shown on figure 1 for Jaccard and Dice, the user will obtain exactly the same results for any of the measures belonging to a given equivalence class.

Another consequence for this field is more profound and concerns the evaluation of the retrieval. For two equivalent measures, any recall/precision comparison based on the lists of the entries retrieved will conclude to the exact same accuracy. In particular, if order is invariant between two lists of results, the question of comparing the numbers of pertinent documents in the first 10 results is void.

Furthermore, as we have pointed it out in section 2.3, if a best-matching procedure uses comparisons of pairs of objects (in our case, regions), the result will be depending only on the order of the resemblance values. If that order is conserved from a measure to another, the final matching will be identical.

5.2 Value-Based Procedures

As we have discovered in section 3, two equivalent measures S_a and S_b are linked by composition of a strictly increasing function f . This extends the result of our discussion to value-based procedures such as thresholding, aggregation, etc. For some purpose beyond the simple query-by-example scheme, we may take into account the value computed by S_a in a specific operation (averaging with some other result, thresholding). In this case, we can predict the outcome of this operation using S_b instead of S_a (every other thing being equal) by using the equivalence function f to predict the value of S_b .

As an example, if we filter our query results by some lower threshold (all results must have a similarity value above 0.5), we know that using S_{dice} (see section 2) rather than $S_{jaccard}$ (histogram intersection) will lead the system to obtain the same ordered results but to present more results because the value of S_{dice} depends on the value of $S_{jaccard}$ and is globally larger.

6 Conclusion

In this paper, we have described both theoretically and empirically existing families of similarity measures that are bound to obtain the same results in a query-by-example perspective. As discussed in the paper, the choice of a similarity measure in this perspective can be reduced to the choice of a family of them. As an example, we have reduced the choice of the parameters of the Tversky's Ratio measures to one unique parameter.

We also discussed the consequences regarding the order of the values issued from a comparison between pairs of objects used in some applications. We have shown that we can practically predict the behavior of a system using one or the other of two equivalent measures.

References

1. C.C. Aggarwal, A. Hinneburg, and D.A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *ICDT 2001 (LNCS 1973)*, pages 420–434, London, UK, 2001.
2. B. Bouchon-Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, 84(2):143–153, 1996.
3. M.D. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, September 1995.
4. A. Hinneburg, C.C. Aggarwal, and D.A. Keim. What is the nearest neighbor in high dimensional spaces? In *26th International Conference on Very Large Data Bases, VLDB2000*, pages 506–516, Cairo, Egypt, September 10-14, 2000.
5. A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 8(29):1233–1244, 1996.
6. J.F. Omhover and M. Detyniecki. Strict: an image retrieval platform for queries based on regional content. In *CIVR'2004 (LNCS 3115)*, pages 473–482, Dublin, Ireland, August, 2004.
7. J.F. Omhover, M. Detyniecki, and B. Bouchon-Meunier. A region-similarity-based image retrieval system. In *IPMU'2004*, pages 1461–1468, Perugia, Italy, July, 2004.
8. J. S. Payne, L. Hepplewhite, and T. J. Stonham. Perceptually based metrics for the evaluation of textural image retrieval methods. In *Int. Conf. on Multimedia Computing and Systems*, volume 2, pages 793–797, Italy, 1999.
9. M. Rifqi, V. Berger, and B. Bouchon-Meunier. Discrimination power of measures of comparison. *Fuzzy Sets and Systems*, 110(2):189–196, March 2000.
10. S. Santini and R. Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, September 1999.
11. M. Stricker and M. Orengo. Similarity of color images. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases*, pages 381–392, San Diego, USA, 1995.
12. M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
13. Y. A. Tolias, S. M. Panas, and L. H. Tsoukalas. Generalized fuzzy indices for similarity matching. *Fuzzy Sets and Systems*, 120(2):255–270, 2001.
14. A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

Developing AMIE: An Adaptive Multimedia Integrated Environment

Osama El Demerdash¹, Sabine Bergler¹, Leila Kosseim¹,
and P. Karen Langshaw²

¹ Concordia University, Department of Computer Science and Software Engineering

² Concordia University, Department of Design and Computation Arts

Abstract. Large multimedia repositories can be used more effectively by providing a hybrid environment for accomplishing common tasks, such as searching, browsing, presenting and indexing of the material. In order to achieve this end, the semantic and relational information must accompany the technical, content-based data. In this paper, we present the framework underlying the development of a hybrid retrieval/presentation environment and a prototype in its early stages of development.

1 Introduction

More and more applications are being developed to process considerable multimedia repositories of images, video, animation, voice, sound and music objects, generally requiring storage, indexing, retrieval and presentation of the media. Earlier research in multimedia dealt with content-based retrieval, the automatic recognition and processing of the content of the medium as in QBIC [1]. However, modeling of the data and task has often been secondary to information retrieval. Temporal and spatial models as well as network issues overshadowed semantic, contextual and relational factors. More recent applications such as EGO [2] started addressing the semantic gap and focusing on user-centered approaches.

We present in this work, our efforts to create a heterogeneous adaptive multimedia environment that effectively supports multiple tasks. We have identified the following goals toward the development of this environment:

1. Maintaining sufficient semantic coherence and cohesion
2. Designing a general context-driven framework
3. Providing just-in-time support to common multimedia tasks including indexing, searching, browsing and presentation

Adaptive Multimedia in Context. For semantic and context modeling of multimedia tasks, we borrow concepts from the fields of discourse analysis and rhetorical structure.

Systemic Functional Linguistics (SF). O'Toole demonstrates through the analysis of a painting [3] that Systemic Functional linguistics [4] is broad enough to

cover other semiotic systems, particularly visual ones. In his analysis, the different constituent functions of the model (ideational, interpersonal and textual) are projected over the representational, modal and compositional functions in the visual domain. In the Systemic Functional model, text is both a product and a process. Language construes context, which in turn produces language [4]. In the light of this theory, it is possible through analysis to go from text to context, or through reasoning about the context to arrive at the text — though not the exact words — through the triggering of the different linguistic functions. While we do not try to draw exact parallels between the Systemic Functional model as applied in linguistics and in multimedia, we retain some of the highlights of this theory; most notably the relation between text — in our case multimedia — and its context and use them to guide us in building a context model for the task.

Rhetorical Structure Theory (RST). We also draw on Rhetorical Structure Theory (RST) [5] for representing the possible relations between the different components of the model. RST has been used to analyze the relations between text spans in discourse and to generate coherent discourse. RST analyzes different rhetorical and semantic relations (ex. precondition, sequence, result) that hold between its basic units (usually propositions). In our framework, we use RST as a design solution to guide us in establishing coherence by modeling the relations among multimedia data similarly to the relations among text spans in a discourse.

2 Related Work

Multimedia applications are task, domain, process or media dependent. Due to the resulting complexity it is necessary for any framework/model to strike a balance between generality and applicability. Jaimes [6] describes a visual information annotation framework for indexing data at multiple levels. The MATN (Multimedia Augmented Transition Network) by Chen et al. [7] proposes a general model for live interactive RTSP (Real-Time Streaming Protocol) presentations, which models the semantics of interaction and presentation processes such as Rewind, Play, Pause, temporal relations and synchronization control (e.g. concurrent, optional, alternative), rather than the semantics of the content. The HIPS project is the most relevant to our discussion since it includes some modeling of the context, the user and their interaction for the case of a museum guide [8] and [9]. A portable electronic museum guide transforms audio data into flexible coherent descriptions of artworks that could vary with the context. The system uses the *MacroNode* approach, which aims to develop a formalism for dynamically constructing audio presentations starting from atomic pieces of voice data (macronodes) typically one paragraph in length. A museum visitor, could get one of several realizations of the description of an artwork depending on the context of interaction. The context is defined according to the visitor's physical location in relation to the described artwork. In this approach, the data is annotated with the description of content and relations to

other nodes. These relations are conceptually similar to relations in Rhetorical Structure Theory. In a later project by Zancanaro [10], the utilization of RST relations is extended to producing video like effects from still images, driven by the audio documentary. A closed-set ontology is employed. Kennedy et al. [11] developed a communicative act planner using techniques from Rhetorical Structure Theory (RST). The purpose of the system is to help animators by applying techniques to communicate information, emotions and intentions to the viewer. The knowledge base of the system includes information about scenes, shots, space, time, solid objects, light, color, cameras and cinematographic effects. The tool is intended to be used in the planning phase to alter a predefined animation in a way perceptible to the viewer. Faceted metadata annotation has been applied to support searching and browsing with promising usability evaluation in [12].

3 An Adaptive Framework

Figure 1 is an illustration of the components of our adaptive multimedia framework, an earlier version of which we presented in [13]. *Static Components* refers to elements not contributing directly to the adaptive potential of the framework, i.e. fixed for different tasks. *Dynamic Components* are responsible for the adaptive aspect of the framework, which is designed to be general enough for different contexts, and complex tasks requiring more elaborate utilization of the framework. A partial proof of concept implementation of this framework will be described in Section 4, including the data model, the information retrieval model and limited areas of the context model, selection heuristics and effects. Here, we present all envisioned alternatives as design solutions.

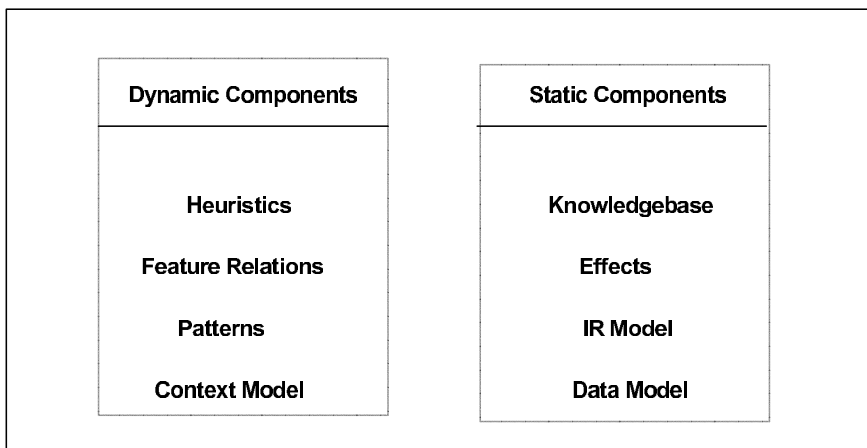


Fig. 1. Framework for Adaptive Multimedia

3.1 The Data Model

Our data model consists of the data files (in any format supported by the visualization software¹) and their annotations with technical and semantic features as well as relational characteristics. Following Prabhakaran [14], we model multimedia objects as a general class with specialized classes for each type of media.

Meta-data describe semantic features of the media files and are constant across media types. These include *Keywords* describing the semantic content and the *Mood* of the selection. Defining these subjective annotations is part of the creative process of performance design and may be more or less useful for another performance. To achieve reusability, general classification ontologies can be used such as distinguishing between *General*, *Abstract* and *Specific*. *General* refers to a class of objects with physical presence like *human*, *chair*, *dog*. *Abstract* is an idea or concept without a physical presence like *hunger*, *war*, *sleep*, and *Specific* is a subclass of *General* for identifiable named entities.

This model is extensible through the use of any relevant ontology, since the annotations are cumulative. For instance, in the current project we include a feature called *Mental Space* with the attributes *dream/reality/metaphoric* and another feature *Physicality* to convey relative size of objects with the attributes *landscape/body/page*. These features may not be relevant to all performances, but may be interesting to some.

Each type of media is also annotated according to its specific characteristics as illustrated in the remainder of this section.

Text. To delimit the medium ‘text’ is a tricky task. Audio data may contain spoken text, images may contain text fragments, complete poems could be laid out in a visual way (as frequently done by Apollinaire), and video and animation may include both spoken and visual text. We define *text* as textual data formatted in ASCII format (e.g. txt, rtf, HTML). Text is the most researched medium in the field of information retrieval and we employ tools similar to many modern search engines, based on prior indexing of keywords.

Image. Still images are digital graphics containing drawings, paintings, photographic images, text or any combination of these. Technical features commonly used for annotating graphics include color, texture, dimensions and file format. In this project texture was deemed irrelevant and excluded. Multiple color annotations were permitted. Sequences of still images are handled using relations between specific files, or through retrieving by patterns, with control over certain features (e.g. speed) through the user interface.

Moving Images. Moving images include videos and animations. We use atomic excerpts consisting generally of a few seconds to two minutes. This roughly corresponds to the definition by [15] of a *Scene*. A *Scene* is a collection of contiguous logically related shots, while shots are contiguous frames with common content. Sequences, which form a higher level in this hierarchy, are not considered as

¹ We currently use Flash-MX which supports formats including mp3, mpg, swf, html...

units, but are dealt with through relations. This category has a time dimension, represented by the *duration* and the *pace* features. The choice of Scene as the basic unit enables the generalization of image features to the video excerpt. For example, *Color* is the dominant color in the scene.

Audio. Audio data is classified as *music*, *speech*, or *other* sound data (e.g. Electro Acoustic, noise etc.). In addition, temporal features like *duration* have to be indicated. For music, we also indicate *pace* (tempo) using qualitative attributes *fast/medium/slow*, and *type* (*melodic*, *harmonic*, *percussive*, *gestural*). Like text data, speech carries information in natural language.

Relations. Relations between the media files are also annotated. As mentioned earlier, we use modified RST-like relations for two reasons. Firstly, to impose temporal constraints on the order of playing these files, in order to insure the production of a coherent and cohesive presentation. Coherence is achieved through the logical temporal and spatial ordering of the different selections of the presentation, while cohesion results from the synchronization of two or more selections. Secondly, to construct a relational navigation map linking the selections according to their sensory and/or semantic links. Multiple relations can be represented, forming a web rather than tree structure, which is customary - though not a requirement - even in the case of text structure [16]. Since RST relations are rhetorical in nature, we augment them with relations for *temporal constraints* (*follow*, *precede*, *simultaneous*) and others that express pure sensory associations (*phonetic*, *visual*).

Figure 2 shows the representation of the relations between media files in RST format.

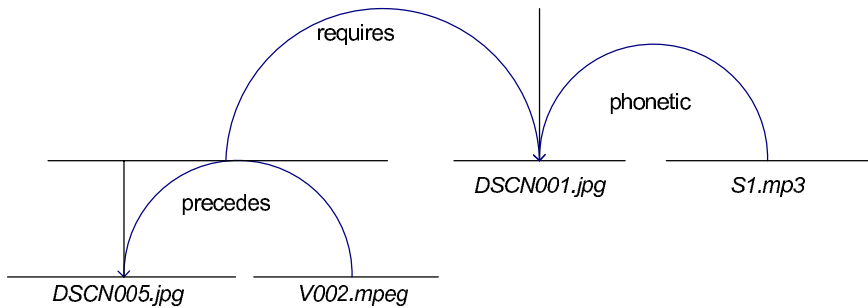


Fig. 2. RST-like relations representation

3.2 The Context Model

For multimedia tasks, we define context to include several interdependent features: the *task*, *outline*, *time*, *space*, *user*, *audience*, *medium*, *rhetorical mode*, *mood*, and *history*. This certainly does not represent a closed set; a more refined framework could of course use several other such features. We give here a brief description of the context variables.

The Task. The tasks intended to be handled by the environment include searching, browsing, presenting and indexing. We have chosen to concentrate in the beginning on the presentation task, since it implicitly includes a retrieval component, and presents more complex requirements than the other tasks. Some of the following sections are only applicable to the presentation task.

The Outline. The outline of a presentation corresponds to the Field of Discourse in SF. It is expressed in terms of keywords. Like the outline of an essay, or a book's table of contents, a presentation outline is a representation of its plan. The user is able to change the subject through the interface of the system, triggering a system response in the form of new material related to the current subject. Outlines could be complex and contain overlapping sections. The outline can be ordered or unordered and must support time constraints as needed, as well as a weighting scheme to indicate the relevance, or relative importance of each keyword in a section. Rhetorical modes and moods for each section should be supplied also.

Time and Space. The intended timeline is a determining factor in the planning of a presentation, since it is used to avoid overflows and empty gaps, as well as to balance media selection. Overflow can happen when a certain media selection, for example a video relating to a particular topic in the outline, turns out longer than that section's initially planned time-slot. Conversely, empty gaps occur when there is not enough material to fill the allotted time for a particular topic. Balancing media selection can help generate more appealing presentations and requires keeping track of time. Time can be modeled at the required level of accuracy (min., sec., etc...).

The physical size of the performance space as well as its placement (inside/outside) provide hints to the appropriate type of media to play. Presets can handle different space configurations.

We define *virtual space* to be the spatial layout on the screen, relevant when multiple objects are presented on the screen simultaneously or when one object does not occupy the full viewing space.

Rhetorical Mode. The rhetorical mode is the communication strategy used at a given moment in a presentation to affect the audience in particular ways. Examples of rhetorical modes given by Halliday include *persuasive*, *expository* and *didactic* [4]. There is no consensus on rhetorical modes in the literature on essay writing, however more modes are usually considered including among others Narrative, Descriptive, Illustrative, Comparison/Contrast, Process analysis, Definition and Cause/Effect.

Moods. The emotional feel of the presentation or its mood contributes to maintaining a coherent context. Moods could either be directly mapped to elements in the taxonomy of the project, or explicit links could be established through the use of feature relations and heuristics. The definition of Moods themselves has to be qualitative and may be comparative (e.g. *happier*, *happy*, *neutral*, *sad*, *sadder*). Color psychology establishes relationships between colors and moods.

For example, Red is often associated with anger and excitement, blue with sadness and calm, green with nature, envy etc. However, other properties of color such as hue and saturation also affect the mood. In music, loudness, rhythm and key are all factors affecting the mood.

Audience. Gender, age, background and relationship to the author of the presentation are all potential selection factors. For example, children might be more responsive to images and animations than to text and video. Artistic, scientific and multidisciplinary audiences require different communicative strategies, for instance for presenting from a position of authority as opposed to a peer-to-peer presentation. Employing stereotypes has become a common practice for modeling anonymous audiences, especially in web-based applications which service a significant number of users with varying characteristics and interests.

The User Profile. Despite their correlations, the user model is often considered separate from the context model in application design. We chose to include the user profile in the context model. Indeed, the user and the audience together correspond to tenor in SF. A user profile can be represented by keywords, preferably drawn from the different ontologies applied in the data model to avoid an extra step of matching terms. Other user profiling techniques include registering the users' tasks and requests to determine their interests.

Media. In the context model, we consider video, audio, animation, image, text and combinations of these, whether simultaneous or overlaid. These media types should not be confused with the medium attribute in the data model, which is used to characterize the medium of single files, or that in the query specification which can be used to constrain the types of media in the result set. The currently playing media types are an essential context parameter and are used by heuristics such as to decide whether or not to interrupt the current selection.

History. A record of selections already retrieved should be used to avoid repetition of these selections. History could also be used to balance, as desired, the concentration of the different media in a presentation and to diversify the selection as required. Moreover, it is possible to use the history to reproduce a presentation, or as training samples for machine learning techniques of context parameters.

3.3 Feature Relations

Relations are used either at the level of individual data files to link selections together as described in the previous section, or at the abstract level. When used as such, they serve to establish explicit relations between the different features of the data model, providing for overriding capabilities, and thus an additional interpretive layer. These relations could be applied within the same medium, for example associating a certain color with a mood, or across different media types, such as yellow with jazz music. Feature Relations can also be used to express constraints, which can be considered as negative relations. For example, to express that Loud music should not accompany Calm mood.

3.4 Heuristics and Experiments

The goal of the selection heuristics is to produce different interpretations of the data, according to the context, and through the selection and ordering of multimedia material. The process involved is a context-to-content mapping. The context of the task, in addition to any explicit triggers, is mapped into specific selections. To refine the relevant heuristics, experiments should be conducted through variations of the different features.

During a given task, the system will keep track of the current context, of changes in goals, and will offer the user to trigger, browse or query the media using a visualization tool convenient for the criteria specified above.

3.5 Visual Effects

Visual effects are techniques used in the presentation model to improve the visual quality of the presentation. They are also used to enhance the relation between two selections in the presentation for example by associating a certain kind of relation with a transition. Effects are applied to alter images, and do not create new ones. They include transitions (*cut*, *fade-in*, *fade-out*, *dissolve*, *wipe*), scaling, zooming, layering etc. These effects are commonly available in the design-mode of presentation software like MS-PowerPoint and Macromedia-Flash, or through programming. However, including them in the run-time interface in an accessible manner, allows the presenter to apply them on the fly during the presentation. The application of visual effects has a long tradition in fields such as cinematography where transitions roughly correspond to punctuation in language.

3.6 Generation Patterns

Patterns are recurring designs, behavior and conditions. In the context of our framework, Patterns could be formed of complex combinations of features and heuristics. Generation patterns are discovered while experimenting with the environment. Once identified and included in the interface, they can be retrieved explicitly during a given task. For example, a Surprise pattern could be a combination of loud dynamics, fast video, and a set of heuristics that changes fast across the different media and colors. This complex goal would be difficult to achieve otherwise in real-time. Defining patterns can also lead to more meaningful ways of describing the higher level goals of the user.

3.7 The Information Retrieval Model

A framework for adaptive multimedia must include an information retrieval component, since pre-arranging all possible combinations of media would be infeasible in large repositories. The information retrieval model defines the way the selection criteria are applied to the annotated data to determine the relevance of documents.

3.8 The Knowledge Base

While some expertise already exists in each medium separately, there is no evidence of standardized practices in the handling of adaptive multimedia tasks. Once the expertise in the domain of multimedia has been developed, it is beneficial to capture this expertise and exploit it in a systematic manner. The Knowledge base would act as a permanent repository of this expertise. Such expertise might include for example techniques, feature relations, heuristics, ontological hierarchies of strategies, meanings, effects and rhetorical relations.

4 AMIE

AMIE (Adaptive Multimedia Integrated Environment) is a prototype for a hybrid multimedia retrieval and presentation environment in its early stages of development. It is developed using a three-tier software architecture on flash/java/MYSQL platforms as illustrated in Figure 3. The model tier consists of the data and annotations, relations and retrieval patterns. Business logic including operations on the database, heuristics and status information makes for the middle tier, while the presentation (view) tier has the user interface and interaction elements. Long-term experimental goals, the volatile nature of requirements and other practical considerations have influenced the three-layered, modularized architecture. Separating the presentation from the model and the business logic permits the substitution of any of these layers at minimum cost.

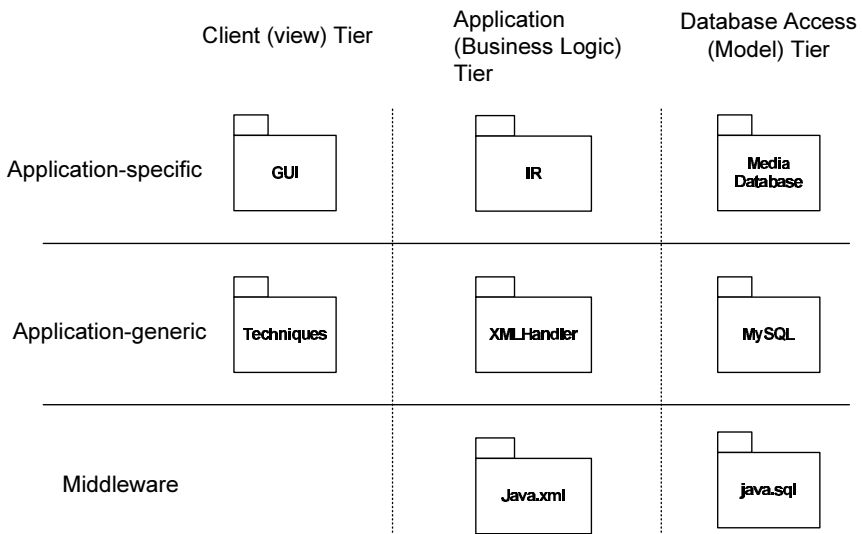


Fig. 3. Architecture of the System

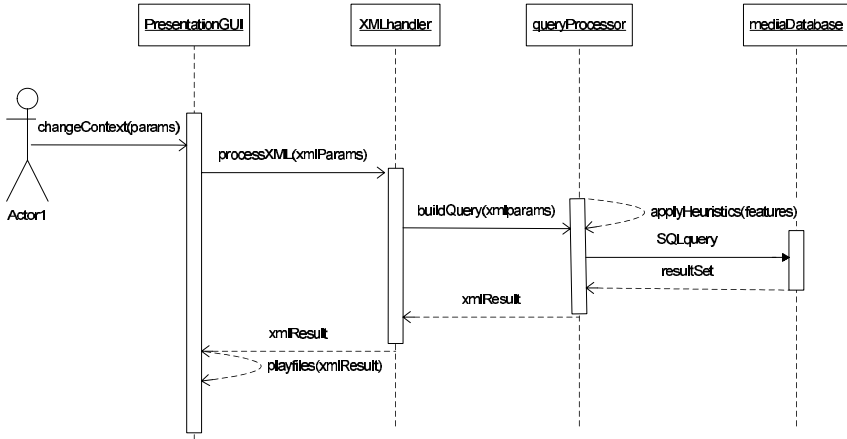


Fig. 4. Sequence Diagram for Retrieving Data

Figure 4 illustrates the interaction sequence for retrieving data , with the following scenario: The user indicates through the presentation graphical user interface (GUI) the features of the data to be retrieved. The presentation GUI reproduces the user’s request in XML format and sends a message to the XML handler to process the request. XML sent by the presentation GUI to the XML handler includes elements for both <sound> (audio) requests and <content> (visual) requests. The XML handler forwards the request to the Query Processor. The Query Processor applies heuristics relevant to the required features. The Query Processor constructs a SQL statement according to the requested features and heuristics and runs it on the media database. The media database returns the result set to the Query Processor. The Query processor translates the result set into XML format and sends it to the XML Handler. The XML Handler forwards the XML result set to the Presentation GUI. The <Slides> element represents visual files while the <Sounds> element represents audio files. It is necessary since the Presentation GUI deals with these categories separately and in different manner. The presentation GUI displays the files specified in the result set.

The interface is used for informing the system of changes in the context model and to control/override the system’s suggestions using relevance feedback and navigation of the multimedia repository.

Figure 5 shows a screen shot of the system. The user can select any of the direct features (*time, spectrum, alpha*) through the presentation GUI. They are either linked internally to the context and data models or generate visual effects. Using the relations (Section 3.3) and the user specifications, the most appropriate data files are retrieved from the multimedia database. Of the relevant data files, only a subset may be used in the final result. The final selection and ordering for the presentation is made by the selection heuristics and the generation patterns which ensure a coherent final presentation.

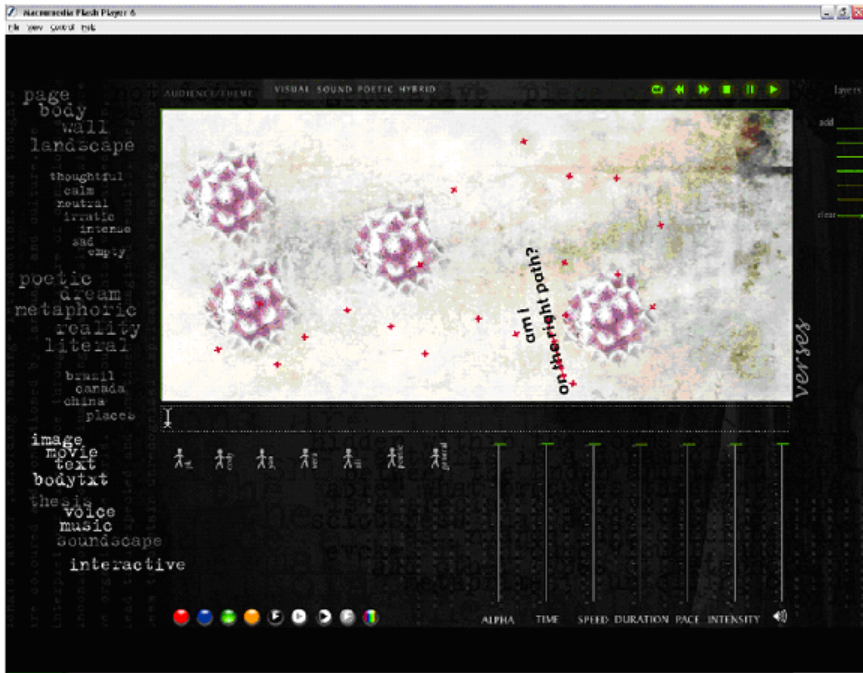


Fig. 5. Screen-shot of the system's interface

5 Evaluation

The presented framework has to be evaluated over time on mostly qualitative reusability measures, such as applicability, scalability and ease of adoption. Our prototype implementation using this framework is a first attempt. Here, we are mainly concerned with identifying and implementing benchmarks to evaluate similar systems. Common methods for evaluating information retrieval systems focus on measuring the effectiveness of the system [17], defined as the relevance of retrieved documents as determined by a human judge. Precision and Recall figures are then calculated for the system.

The influential Text Retrieval Conference (TREC) adopts this benchmark. However, [18] questions this measure, pointing out that Precision and Recall measures ignore such important factors as the relativity of the relevance of a document. The binary division into relevant and non-relevant documents is an oversimplification. [19] notes, Precision and Recall presume that an objective judgment of relevance is possible, an assumption readily challenged by the high interannotator discrepancies.

The case of Multimedia Information Retrieval offers other particularities and difficulties which need to be considered for evaluation. The TREC 2001 Proceedings, which included a video track for the first time, acknowledge the need for a different evaluation system for that track [20]. [18] lists some of the

performance criteria not captured by Precision and Recall are speed of response, query formulation abilities and limitations and the quality of the result. He presents the notion of Approximate Retrieval (AR) arguing that unlike text data, the characteristic ambiguity of both multimedia information and queries could only lead to an approximate result, and asserts the significance of rank, order, spread and displacement in Multimedia Information Retrieval. Schauble [17] introduces a notion of subjective relevance which hinges on the user and her information needs rather than on the query formulation. Thus we feel that the user's participation in determining the relevance of the result is an essential factor.

Certain characteristics of our system add to the complexity of the evaluation task. These include the user model, context-sensitive retrieval, and the layer of subjective relations between features and/or elements in the data model introduced explicitly by the user.

Alternative user oriented measures for the evaluation of the system have been suggested. The first two of these measures, namely the Coverage and Novelty Ratios reported by [19] measure the effectiveness of the system with respect to the user's expectations. The Coverage Ratio measures the ratio of documents which the user was expecting to be retrieved over what was actually retrieved by the system:

$$\text{Coverage} = \frac{\# \text{ of relevant documents known to the user and retrieved}}{\# \text{ of relevant documents known to the user}},$$

while the Novelty Ratio is the ratio of relevant documents which were not expected by the user:

$$\text{Novelty} = \frac{\# \text{ of relevant documents retrieved previously unknown to the user}}{\text{total } \# \text{ of relevant documents}}.$$

In order to apply these measures, a special evaluation environment needs to be set up to run the system in interrupted mode so that the user can evaluate the selections played without interfering with the system heuristics.

Questionnaires could help evaluate the system from a different perspective: its higher-level goals of providing the user with an effective tool and an enjoyable experience. Such user-oriented methods have been applied by Jose in [21].

6 Conclusion and Future Work

Multimedia provides a powerful tool for communication. However, in order to exploit the full potential of multimedia, it is essential to allow for a certain flexibility in task handling. We proposed a framework for a hybrid adaptive multimedia integrated environment. The framework included models for data, context and retrieval, selection heuristics, retrieval patterns and multimedia techniques. As a proof of concept, we implemented a prototype that dynamically selects and plays the most appropriate selection of multimedia files according to the preferences and constraints indicated by the user within the framework. We borrowed concepts from text analysis to model the semantic dimension of the environment. We also proposed adopting methods of evaluation which reflect the subjective nature of the tasks.

We intend to expand the environment to include tools for accomplishing common tasks such as data annotation and interface design. The goal is a robust architecture, with acceptable scaling and generalization capacity. Some areas where we see possibilities for improvements and innovation are:

- Using other triggering mechanisms as speech, gesture, multi-channel, multi-modal and cluster-based content navigation maps to navigate through the concept space and visualize the relations between the concepts
- Developing an annotation tool to help the inexperienced user
- Investigating alternative architectures like agent-based architectures
- A web-based multi-user version would allow for collaborative task accomplishment in real-time.
- Using supervised machine-learning techniques and relevance feedback for discovering heuristics and building user profiles through the collection of history data

References

1. Flickner, M., Sawhney, H., Nublack, W.: Query by Image and Video Content: The QBIC System. In: Intelligent Multimedia Information Retrieval. California: AAAI Press/ The MIT Press (1997)
2. Urban, J., Jose, J.M.: EGO: A personalised multimedia management tool. In: Proc. of the Second International Workshop on Adaptive Multimedia Retrieval. (2004)
3. O’Toole, M.: A Systemic-Functional Semiotics of Art. In: Discourse in Society: Systemic Functional Perspectives. Ablex Publishing Corporation, New Jersey (1995)
4. Halliday, M., Hasan, R.: Context and Text: Aspects of Language in a Social Semiotic Perspective. Oxford University Press, Oxford (1989)
5. Mann, W., Matthiessen, C., Thompson, S.: Rhetorical Structure Theory and Text Analysis. In: Discourse Description: Diverse Linguistic Analyses of a Fund-raising text. John Benjamins Publishing Company, Amsterdam (1992) 39–78
6. Jaimes, A., Shih-Fu: A conceptual framework for indexing visual information at multiple levels. In: SPIE Internet Imaging 2000. Volume 3964. (2000) 2–15
7. Chen, S.C., Li, S.T., Shyu, M.L., Zhan, C., Zhang, C.: A multimedia semantic model for RTSP-based multimedia presentation systems. In: Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering (MSE2002), Newport Beach, California, ACM (2002) 124–131
8. Not, E., Zancanaro, M.: The MacroNode approach: Mediating between adaptive and dynamic hypermedia. In: Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems. (2000)
9. Marti, P., Rizzo, A., Petroni, L., Diligenti, G.T.M.: Adapting the museum: a non-intrusive user modeling approach. In Kay, J., ed.: User Modeling: Proceedings of the Seventh International Conference, UM99, Banff, Canada, Springer Wien New York (1999) 311–313
10. Zancanaro, M., Stock, O., Alfaro, I.: Using cinematic techniques in a multimedia museum guide. In: Proceedings of Museums and the Web 2003, Charlotte, North Carolina, Archives and Museum Informatics (2003)

11. Kennedy, K., Mercer, R.: Using communicative acts to plan the cinematographic structure of animations. In Cohen, R., et al., eds.: *Advances in Artificial Intelligence: Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence, AI2002*, Calgary, May 27-29, Springer, Berlin (2002) 133–146
12. Yee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: *CHI '03: Proceedings of the conference on Human factors in computing systems*, ACM Press (2003) 401–408
13. El Demerdash, O., Langshaw, P., Kosseim, L.: Toward the production of adaptive multimedia presentations. In Thwaites, H., ed.: *Ninth International Conference on Virtual Systems and Multimedia - Hybrid Reality: Art, Technology and the Human Factor*, Montreal, International Society on Virtual Systems and Multimedia VSMM (2003) 428–436
14. Prabhakaran, B.: *Multimedia Database Management Systems*. Kluwer Academic Publishers (1997)
15. Carrer, M., Ligresti, L., Ahanger, G., Little, T.D.: An Annotation Engine for Supporting Video Database Population. In: *Multimedia Technologies and Applications for the 21st Century: Visions of World Experts*. Kluwer Academic Publishers (1998) 161–184
16. Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA (2000)
17. Schäuble, P.: *Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases*. Kluwer Academic Publishers (1997)
18. Narasimhalu, A.D., Kankanhalli, M.S., Wu, J.: Benchmarking Multimedia Databases. In: *Multimedia Technologies and Applications for the 21st Century: Visions of World Experts*. Kluwer Academic Publishers (1998) 127–148
19. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. ACM Press and Addison Wesley (1999)
20. Smeaton, A.: The TREC 2001 video track report. In Voorhees, E., Harman, D., eds.: *The Tenth Text Retrieval Conference, TREC 2001*. NIST Special Publication 500-250, NIST, Gaithersburg, Maryland (2001) 52–60
21. Jose, J.M., Furner, J., Harper, D.J.: Spatial querying for image retrieval: a user-oriented evaluation. In: *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press (1998) 232–240

Exploring the Structure of Media Stream Interactions for Multimedia Browsing

Saturnino Luz and Matt-Mouley Bouamrane

Department of Computer Science,
Trinity College Dublin, Ireland
{luzs, bouamram}@cs.tcd.ie

Abstract. This paper presents an approach to the issue of adding structure to recordings of collaborative meetings supported by an audio channel and a shared text editor. The virtual meeting environment used is capable of capturing and broadcasting speech, gestures and editing operations in real-time, so recording results in continuous multimedia data. We describe the implementation of a browser which explores simple linkage patterns between these media to support information retrieval through non-linear browsing, and discuss audio segmentation issues arising from this approach.

1 Introduction

It is generally accepted that providing effective access to time-based data is the most challenging aspect of multimedia information retrieval. Unlike text, a medium for which there is a large and well established body of information retrieval methodology [1], time-based media recordings such as audio and video cannot be straightforwardly mapped to verbal descriptions, remaining the subject of intense research activity. Although there appears to be no definite paradigm in this area, standards such as MPEG-7 [2, 3] are starting to emerge which aim at adding descriptive annotation and structure to multimedia data. Since manual annotation is mostly impractical given the large and increasing volumes of time-based recordings available, automated solutions have been sought.

A great deal of research in automatic indexing of and access to time-based media has focused on the issue of *translating* these media from their typical transient and sequential form into a parallel and persistent presentation [4, 5, 6]. Descriptors thus extracted can then be used in conjunction with existing text retrieval techniques to provide content-based indexing. This approach builds on the structuring role time naturally plays in multimedia data, whereby content extracted from the audio track through speech recognition, for instance, could provide valuable links to video content. Due to the continuous nature of these media, visualisation and retrieval interfaces based on this approach tend to emphasise linear access (whether sequential or random), often employing a “tape recorder metaphor” and building upon it a set of media and domain specific

improvements, such as skimming [7], parsing with compressed data [8] and summary generation [6].

In many cases, such as recordings of lectures, television broadcasts and meetings, time-based media encompass static data which become bound by the same temporal constraints as continuous data. If the components of such multimedia recordings can be treated as separate *timed streams* [9], text and graphics can help uncover and provide access points to non-linear (e.g. thematic) relationships in time-based media. Although the study of such relationships has received far less attention than modality translation in multimedia information retrieval research, we argue that a systematic approach to the former can complement and improve the functionality provided by the latter.

In this paper, we focus on mappings between text and audio streams in recordings of audio conferencing meetings. The recordings used were captured from a system that allows real-time editing of shared documents supported by an audio channel [10]. They contain “video-like” features, in the sense that it is possible to play back all actions involved in editing, including gesturing and awareness feedback (e.g. shared scrollbar motion).

Despite the fact that concurrent text and audio media complement each other effectively in live meetings, post-meeting “salvaging” of information [11] from recorded audio and text data is often a challenging task. It has been suggested that providing users with efficient and simple interfaces for browsing, retrieving and manipulating meeting content might be the key to an increased use existing on-line conferencing technologies. We describe work in progress on a content representation and browsing tool for remote, computer-mediated, synchronous collaborative writing supported by a speech channel, which illustrates a modality of non-linear access to time-based media based solely on the analysis of patterns of interaction between streams. We believe this scenario represents a paradigmatic case of supporting access to the *environmental context* [12] in multimedia retrieval.

2 The Meeting Recording Environment

The meetings targeted by our prototype are non-located *speech-and-text* collaborative activities as described in [13]. Speech-and-text meetings typically involve a small number of participants. Recordings of such meetings were produced using a combination of existing MBone multi-conferencing technology, such as UCL’s real-time audio conferencing tool [14], and a meeting recorder and collaborative text editor implemented specifically for the purpose of collecting data for this research [10].

After some post-processing, the meeting recorder produces an archive consisting of: a searchable image of audio real-time protocol (RTP) packets exchanged among the various clients (including the original synchronisation and source data), decoded audio files, a profile of user activity detailing the time of individual speech exchanges, gesturing (as supported by the collaborative editor’s awareness widgets) and editing operations, and an XML file containing the

```

[...]
<segment id="11">
  <timestamp actionid="1" agent="1" action="insert" start="493" end="493" />
  <timestamp actionid="49" agent="2" action="gesture" start="3550" end="3550" />
  <timestamp actionid="50" agent="2" action="gesture" start="3553" end="3553" />
  <timestamp actionid="55" agent="2" action="delete" start="3602" end="3602" />
  <timestamp actionid="110" agent="2" action="delete" start="4273" end="4273" />
  <timestamp actionid="127" agent="2" action="delete" start="4721" end="4721" />
  <timestamp actionid="129" agent="2" action="insert" start="4722" end="4723" />
  The first thing people asked was what about the Blue Jay. Had he stayed blue? Yes, he was
  still the same color. No longer were there two colors in the world, but just one -- the
  color blue. And because the Blue Jay was a color like everybody and everything else people
  began to lose interest. Now that he was neither more nor less important crowds stopped
  coming and one day, six months into the year that the world had turned blue, somebody let
  him out of his cage and he flew off looking happy to be free.
</segment>
<segment id="12.1">
  <timestamp actionid="105" agent="1" action="insert" start="4188" end="4188" />
  <timestamp actionid="131" agent="2" action="delete" start="4729" end="4729" />
  But on occasion they wondered where the Blue Jay had gone and how he was doing and, most
  of all, if he was still the color blue and what it had all meant.
</segment>
[...]
<action id="50" segments="11" startPar="11" points="(189,26)">The</action>
[...]
<action id="110" segments="11,12" startOffset="0" endOffset="0">
</action>[...]

```

Fig. 1. Excerpt of XML file showing representation of segments, timestamps and actions in a typical meeting

textual content of the shared document along with interaction metadata. The metadata contained in this XML file describe, for each text segment, a list of timestamps of operations performed on that segment. The timestamps record the name of the agent that performed the operation, the start and end time of the operation, and the nature of the operation — e.g. *Edit*, *Insert*, *Point*, and *Gesture*. Timestamps are complemented by detailed action description metadata stored in the same file. The recorded operations are visible to all participants in real time during the meeting.

A typical fragment of XML-formatted text is shown in Figure 1. It illustrates how timestamp tags are attached directly to text segments (top) while action descriptions are appended to a separate section at the bottom of the file. As the text evolves, action tags might lose their links to their original timestamps. That would occur, for instance, if a text segment were moved through a cut-and-paste operation, since moving implies assignment of value to segment `id` attribute, or completely deleted from the document. Action descriptions are, however, detailed enough to allow the document to be reconstructed step by step [10].

3 Non-linear Access

One can regard the communication modalities involved in speech-and-text meetings as creating two orthogonal dimensions. If one simply considers the textual outcome of the meeting, one ignores the process which led to this final outcome.

On the other hand, linear access through, for instance, listening to a whole meeting while watching the evolution of the document is time-consuming and clearly unsatisfactory from an information retrieval perspective.

A better strategy would be to start the search through text, say, by selecting certain topics, and then play, in turns, those speech segments which relate to the selected text segments in order to contextualise the information they convey. Since participants often refer back to text segments in speech while pointing at them, the user also has to be able to visualise these text referents as needed. Consider, for instance, the following (collaboratively written) text fragments extracted from a corpus of student-supervisor meetings [15]:

- (t1) *Is the mobile visualization an improvement over a simple text based itinerary? (simple conventional paper-base) (clarify what is being compared!)*
- (t2) *(also find out if general user preference exist as far as a number of interface options i. do they like clocks on turning points or on the line; [...])*

The first segment (t1) is *related* (as explained below) to six audio segments. In one of those audio segments (s3), the speaker makes an elliptical reference to the second text segment (t2), as shown in the following transcribed speech fragment:

- (s3) *ok [pause] because [pause] yes, to start with you're saying that over here [point at (t1)] but as you get down to these parts [point at (t2)] you are talking about text-based interfaces on the mobile phone versus the graphical form on the phone [...]*

Text segment (t2), by its turn, is linked to another seven audio segments, each of which might be linked to a number of text segments, and so on. Such linking

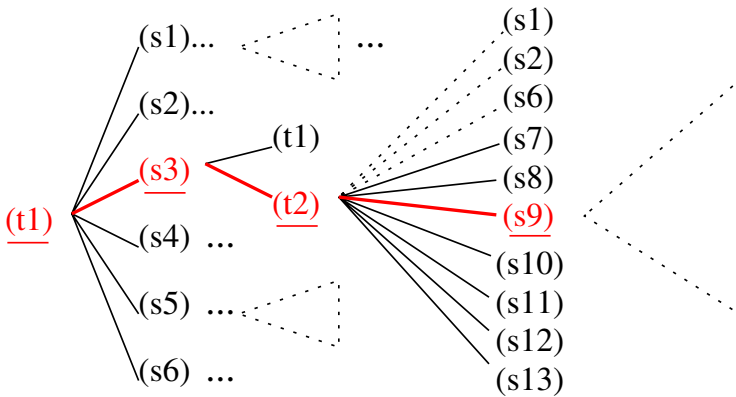


Fig. 2. Sample navigation structure of alternating speech (s1, ..., s13) and text (t1, t2) segments

patterns can be represented by tree structures, as illustrated in Figure 2. This kind of navigation structure can evolve along several dimensions, beyond what is supported by meeting browsers based on a linear access metaphor, including those supported by sophisticated modality translation components [16, 17].

In the following section we describe an approach to browsing that shifts the emphasis from *time* to *objects* (speech and text segments) and *actions* (editing operations and gestures), and define structures based on the relationship between these media. A visualisation prototype is presented which highlights non-linear and hierarchical aspects of speech-and-text meeting recordings.

4 The MeetingTree Prototype

As an alternative to representing a meeting along a time-line, we propose a mapping where concurrency of speech and editing operations determines links between text and audio segments. We start with the assumption that text and speech can be clustered into natural segments, and define two types of *meeting objects*: *speech segments* and *text segments*. For simplicity, a speech segment can be initially defined as an individual audio interval delimited by silence. Similarly, a text segment can initially be defined as a paragraph, or chunks of text delimited by two sets of consecutive line breaks. We discuss issues relating to segmentation in more detail in section 4.3.

There are many ways to define contextual and temporal relationships between these meeting units as well as several possible starting points [18]. We have chosen to build a representation of the conference starting from the final form of the textual outcome of the meeting. This provides a natural and intuitive starting point from which to explore how this outcome was attained.

The timestamp elements in the XML document support extraction of all operations performed on any text segment. Using the time intervals contained in these timestamps, we are able to retrieve all speech intervals overlapping with a single text operation. Hence, each segment can be linked with speech segments which occurred while the paragraph was created, edited, modified or pointed at. We define the relationship between a speech segment and a paragraph as a “speech-while-text” link. This relationship was chosen on the intuitive assumption that in collaborative writing, what is being said during modification of some part of the text is likely to be relevant to the content of the text itself. The speech segments are arbitrarily long, so in some cases, in the duration of a particular speech segment, some other text operation might have been performed on another paragraph. This therefore enables us to extend the links by defining the “text-while-speech” relationship. Once again, this relationship was chosen on the assumption that, if some text is modified while participants are speaking, these text modifications are likely to be somehow related to what is being currently said.

These relations are formally defined in [18] in terms of *temporal neighbourhoods*. Given a set $T = \{t_1, \dots, t_{|T|}\}$ of text segments, and a set of speech segments $S = \{s_1, \dots, s_{|S|}\}$, temporal neighbourhoods are determined by interval overlap as shown below.

Definition 1. A temporal text-audio mapping is a function $tn : T \rightarrow 2^S$ so that $tn(t_i) = \{s_j : t_i^s \leq s_j^s \leq t_i^e \vee t_i^s \leq s_j^e \leq t_i^e \vee s_j^s \leq t_i^s \leq s_j^e \vee s_j^e \leq t_i^e \vee s_j^s \leq t_i^e \leq s_j^e\}$, where t_i^s and t_i^e denote the start and end time of segment t_i .

Definition 1 characterises the temporal mapping so as to include all relations between text and audio segments allowed in Allen’s time interval algebra [19] except for the *before* relation and its converse. This definition is somewhat arbitrary in the sense that it does not attempt to capture finer grained distinctions in the space of all possible temporal relations between the two segmentations. It is conceivable that some segment relations might be indicative of stronger semantic relations others. An audio segment related to a text segment by a long *overlap* might, for instance, be regarded as more relevant with respect to the content of the text than another audio segment related to the text by a *meet* relation. An empirical investigation of the space of temporal relations along the lines of the strategy proposed in [20] might be useful in this context.

Once tn has been constructed, one can also retrieve specific text segments using audio as a starting point by simply inverting the mapping, or defining an *audio-text mapping* $tn_a : S \rightarrow 2^T$, such that:

$$tn_a(s_i) = \{t_j : s_i \in tn(t_j)\} \quad (1)$$

The relation $\mathcal{T} \subseteq S \times T$ induced by tn is what we call a temporal neighbourhood. Variants of these definitions have been discussed in [13]. The system described in this paper, however, focuses solely on easily identifiable temporal structures defined by the above stated constraints.

Using the above relationships recursively, we have implemented a tree representation of the conference with the text document as root. The paragraphs (text segments) are the first layer of child nodes. Using time interval concurrency, successive layers of speech and text nodes are generated. Each branch of the tree can therefore be viewed as a *temporal neighbourhood*, including all concurrent text operations and speech exchanges. A simple criterion is used for pruning the tree: a single node cannot appear twice on the same branch. Considering that each branch only links nodes in given time intervals, this criterion was sufficient to prune conference trees to appropriate heights.

4.1 Browsing a Meeting with MeetingTree

At its current stage, MeetingTree is an early prototype which has been designed to explore the hierarchical tree structure described above. It has mainly been used as a tool for analysis of meeting data by researchers. We envisage future versions of MeetingTree for users who have attended a meeting and wish to review certain aspects of the meeting. A typical scenario would be organising a work plan. In this scenario, it is likely that only a few sentences would be written down on the editor during the actual meeting due to lack of time. At a later time, MeetingTree would permit users to listen to what was said during the meeting and add to the notes accordingly. It could also be used by a person who was not able to attend the meeting and is curious to know what led to the final outcome.

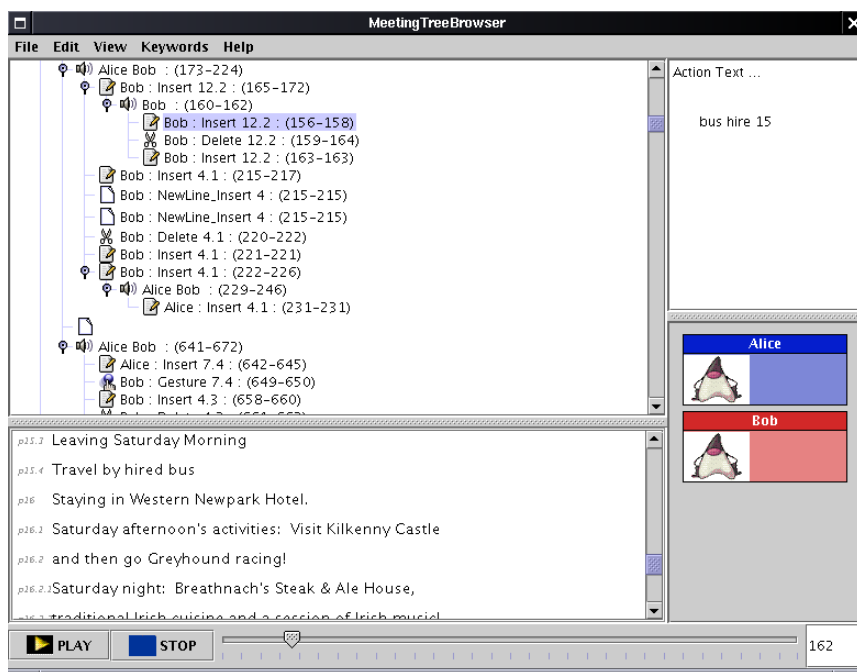


Fig. 3. MeetingTree Browser

Figure 3 shows the MeetingTree browser user interface. The system presents a text document as a list of paragraphs nodes. When the user selects a text segment identifier, its content appears on the lower pane. Expanding the paragraph node will present the user with a layer of audio nodes (if any), each detailing agent and time of speech. By selecting, an audio node, the user can listen to the particular speech interval. If there are further text links (i.e. a different editing operation happened during the speech interval), then the audio node can be expanded, revealing further text nodes. These might detail operations on one or several segments. Selecting such nodes causes the browser to show all segments affected by the editing operation.

In cases where the shared document was uploaded at the beginning of the meeting, using the MeetingTree prototype a user will immediately know at a glance which segments were not modified during the meeting, since these segments will produce no links. On the other hand, a segment displaying many links will have been the subject of various modifications and discussions. Figure 4 illustrates how a user might navigate through a recorded meeting using MeetingTree. The meeting is between two people organising a work plan.

4.2 Semantic Linkage

Even though the MeetingTree prototype only really maps concurrency of text and speech operations, trials of the prototype have shown in many cases a strong

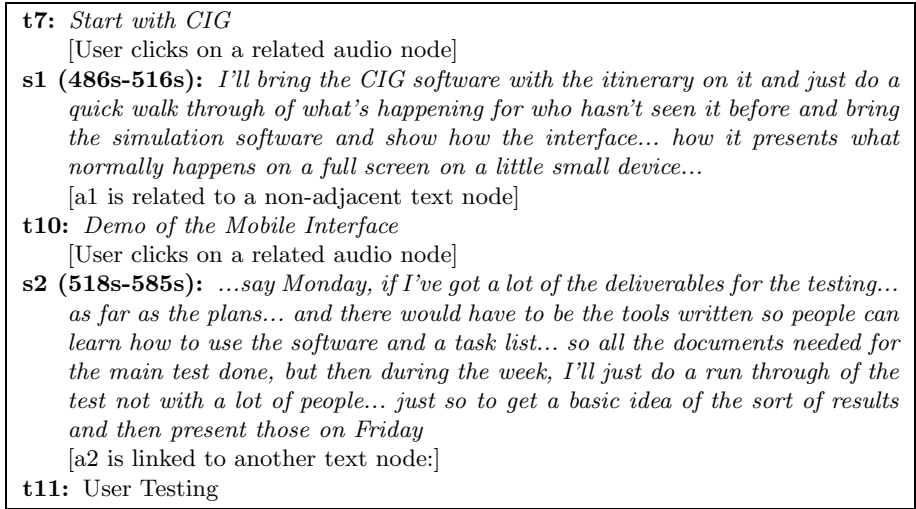


Fig. 4. Example of user interaction with MeetingTree

semantic relationship between the various audio and text segments linked, even after several layers. This aspect has been improved by the reduction of short audio segment mapped on the tree, as discussed in section 4.3. One of the most interesting aspects of the MeetingTree prototype is that it highlights links between non-contiguous text segments, related through common speech segments or editing operations. This is illustrated in Figure 4. The content linking shown there is neither sequential nor can it be derived from a measure of similarity as provided, for instance, by a keyword-spotting approach. Yet, it can offer some insight as to how the participants ideas naturally evolved during the meeting.

4.3 Combining Speech Segments

RTP traffic monitoring or standard endpoint detection techniques can be used for segmenting speech for use by MeetingTree. These techniques, however, tend to yield very large numbers of silence-delimited segments. While the texts written using the shared editor in our corpora are generally short, typically between 20 to 40 paragraphs, there are literally hundreds of speech segments. Showing all these segments on MeetingTree as separate audio nodes would limit the usefulness of the tool as it is not possible to have a full view of the meeting tree on the screen. This is a classic problem for visualisation of large volumes of data, and in audio browsing in particular [5]. We are currently exploring various strategies for reducing the number of speech segments without compromising the integrity and structure of the recording. Taking a closer look at the speech segments in our corpus we realised that a very high proportion of these segments were of very short duration, typically a few seconds. Table 1 shows that in a sample of four meetings, the total number of audio segments of less than 5 seconds account for more than half the total number of audio segments in the meetings.

Table 1. Meeting audio segment statistics

Duration (in s)	No. of segments	< 4s	< 5s
2224	292	149	175
3379	550	343	372
601	55	26	30
1378	143	73	78

In most cases, these short audio segments are mumbles, false-starts, backchannel responses, or short sentences such as a “yes” or a “no”, which are a natural part of any discussion. It could be argued that their occurrence during a remote audio meeting is even higher. Since participants do not meet face to face, they feel a more pressing need to show attentiveness or simply acknowledge each other’s presence in the virtual environment, by means of some sort of verbal feedback. This can be viewed as an aspect of *awareness maintenance* in computer-mediated meetings [21].

The high number of this type of short segments naturally makes them prime candidates for reduction. Of course, the difference between a positive and a negative answer, for example, can be of capital importance in certain circumstances, which rules out ignoring certain audio segments based solely on their duration. While we do not deny the potential semantic significance of these short segments in a recording, we argue that they are semantically poor if taken out of context. Therefore, one of the strategies we have implemented for audio segment reduction consists of merging short segments within larger ones. Merging happens if and only if a shorter segment is totally included in a larger one.

Table 2. Number of audio segments included in a larger segment

Duration (in s)	No. of segments	< 4s	< 5s
2224	292	90	91
3379	550	192	195
601	55	17	19
1378	143	48	49

Although this strategy still leaves us with a large number of speech segments, as seen in Table 2, its implementation results in a significant reduction of the size of the tree displayed by MeetingTree. No information loss is incurred as a result. This, in effect, means that short segments are not individually mapped but appear in context within larger segments. In other words, if a participant said “yes” while someone else was speaking, this utterance will be heard on the audio output, but only the larger speech interval is mapped the visualisation component. Informal user trials of MeetingTree after segment reduction have shown a that the prototype improves performance on the browsing task, compared to simple audio playback combined with (an unstructured version of) the

final text. A detailed experiment has been planned which will further investigate the effectiveness of the system.

MeetingTree also allows the possibility of reducing the number of speech nodes by “deforming” Allen’s temporal relations *before* and *after* between segments, i.e. by regarding segments as semi-intervals [22]. Two segments will be merged if they are separated by an interval whose length falls below a certain *individuation threshold*, calculated as a proportion of the durations of the candidate segments. Although this approach can further reduce the size of the graph, it can also result in a more arbitrary segmentation. We are currently investigating the effects of different merging strategies on the final meeting tree with respect to relevance.

General speech turn detection and discourse structure analysis are complex, open research problems in natural language processing. MeetingTree offers a simple alternative which might help users take advantage of turn taking structure for browsing and retrieving information from recordings of speech-and-text meetings.

5 Conclusions and Further Work

Linking text operations and speech exchanges based on concurrency offers an intuitive, simple and efficient framework for browsing and visualisation of multimedia meetings encompassing two communication modalities. The distinctive feature of the approach described in this paper is that it supports a mode of non-linear browsing which can reveal semantic relationships between non-contiguous meeting segments. This non-linear access mode has also been explored, in a different form, for multimedia browsing on mobile devices [15] with promising results.

Future research will explore different visualisation techniques using a significantly larger corpus of recorded meetings. The prototype described above provides a basis for exploration of recurring patterns in multimedia meeting recordings. We are currently investigating the use of these techniques for communication turn detection and automatic meeting summarisation.

Acknowledgments

This work has been supported by Enterprise Ireland through a Basic Research Grant. The authors wish to thank the AMR '05 reviewers for their comments and suggestions.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley-Longman (1999)
2. Martinez, J., Koenen, R., Pereira, F.: MPEG-7: the generic multimedia content description standard, part 1. IEEE Multimedia **9** (2002) 78–87

3. Martinez, J.: Overview of MPEG-7 description tools, part 2. *IEEE Multimedia* **9** (2002) 83–93
4. Kazman, R., Al-Halimi, R., Hunt, W., Mantey, M.: Four paradigms for indexing video conferences. *IEEE Multimedia* (1996) 63–73
5. Foote, J.: An overview of audio information retrieval. *Multimedia Systems* **7** (1999) 2–10
6. Smeaton, A.F.: Indexing, browsing, and searching of digital video and digital audio information. In Agosti, M., Crestani, F., Pasi, G., eds.: 3rd European Summer School on Information Retrieval. Volume 1980 of *Lecture Notes in Computer Science.*, Springer-Verlag (2001) 93–110
7. Arons, B.: SpeechSkimmer: Interactively skimming recorded speech. In: *Proceedings of UIST'93: ACM Symposium on User Interface Software Technology*, Atlanta, ACM Press (1993) 187–196
8. Zhang, H., Low, C., Smoliar, S.: Video parsing and browsing using compressed data. *Multimedia Tools and Applications* **1** (1995) 89–111
9. Gibbs, S., Breiteneder, C., Tschritzis, D.: Data modeling of time-based media. *ACM SIGMOD Record* **23** (1994) 91–102
10. Bouamrane, M.M., King, D., Luz, S., Masoodian, M.: A framework for collaborative writing with recording and post-meeting retrieval capabilities. *IEEE Distributed Systems Online* (2004) Special issue on the 6th International Workshop on Collaborative Editing Systems.
11. Moran, T.P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., van Melle, W., Zellweger, P.: “I’ll get that off the audio”: A case study of salvaging multimedia meeting records. In: *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*. Volume 1. (1997) 202–209
12. Marchionini, G.: Leveraging context for adaptive multimedia retrieval: a matter of control. In: *Proceedings of The 3rd International Workshop on Adaptive Multimedia Retrieval* (this volume). Springer Verlag (2005)
13. Luz, S., Masoodian, M.: A model for meeting content storage and retrieval. In Chen, Y.P.P., ed.: *11th International Conference on Multi-Media Modeling (MMM 2005)*, Melbourne, Australia, IEEE Computer Society (2005) 392–398
14. UCL Network and Multimedia Research Group: RAT: Real-time audio tool. <http://www-mice.cs.ucl.ac.uk/multimedia> (2004)
15. Luz, S., Masoodian, M.: A mobile system for non-linear access to time-based data. In: *Proceedings of Advanced Visual Interfaces AVI'04*, ACM Press (2004) 454–457
16. Waibel, A., Brett, M., Metze, F., Ries, K., Schaaf, T., Schultz, T., Soltau, H., Yu, H., Zechner, K.: Advances in automatic meeting record creation and access. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. (2001)
17. Morgan, N., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Janin, A., Pfau, T., Shriberg, E., Stolcke, A.: The meeting project at ICSI. In: *Procs. of Human Language Technologies Conference*, San Diego (2001)
18. Masoodian, M., Luz, S.: COMAP: A content mapper for audio-mediated collaborative writing. In Smith, M.J., Savendy, G., Harris, D., Koubek, R.J., eds.: *Usability Evaluation and Interface Design*. Volume 1 of *Proceedings of HCI International 2001.*, New Orleans, LA, USA, Lawrence Erlbaum (2001) 208–212
19. Allen, J.F.: Towards a general theory of action and time. *Artificial Intelligence* **23** (1984) 225–255

20. Ibrahim, Z.A.A., Ferrane, I., Joly, P.: Temporal relation analysis in audiovisual documents for complementary descriptive information. In: Proceedings of The 3rd International Workshop on Adaptive Multimedia Retrieval (this volume). Springer Verlag (2005)
21. Dourish, P., Bellotti, V.: Awareness and coordination in shared workspaces. In: Procs. of the Conference on Computer-Supported Cooperative Work, Toronto, ACM Press (1992) 107–114
22. Freska, C.: Temporal reasoning based on semi-intervals. *Artificial Intelligence* **54** (1992) 199–227

CARSA - An Architecture for the Development of Context Adaptive Retrieval Systems

Korinna Bade, Ernesto W. De Luca,
Andreas Nürnberger, and Sebastian Stober

Information Retrieval Group,
Institute for Knowledge and Language Engineering,
Otto-von-Guericke-University of Magdeburg,
Universitätsplatz 2, D-39106 Magdeburg, Germany
{bade, deluca, nuernb, stober}@iws.cs.uni-magdeburg.de

Abstract. Searching the Web and other local resources has become an every day task for almost everybody. However, the currently available tools for searching still provide only very limited support with respect to categorization and visualization of search results as well as personalization. In this paper, we present a system for searching that can be used by an end user and also by researchers in order to develop and evaluate a variety of methods to support a user in searching. The CARSA system provides a very flexible architecture based on web services and XML. This includes the use of different search engines, categorization methods, visualization techniques, and user interfaces. The user has complete control about the features used. This system therefore provides a platform for evaluating the usefulness of different retrieval support methods and their combination.

1 Motivation

Searching the Web has become an every day task for almost everybody, at work as well as at home. Nevertheless, it is still difficult to find specific knowledge and requires some expertise and intuition in query formulation. One of our main objectives is to make this process a little easier.

Currently, multiple algorithms and systems are being developed for this purpose from many researchers. With CARSA (Context Adaptive Retrieval System Architecture), we want to provide a flexible architecture that allows for experimenting with different methods and their combination in a real world retrieval setting. Various methods for searching, classification, clustering, ranking and visualization can be easily integrated and combined to examine their effects on improving web and local search. We focus especially on the integration of methods that support the adaptation of the system interface and the output to the current search context. This includes methods that consider the *semantic search context* based on search results and user interests, but also the *physical search context* defined, e.g., by the device and its interaction components a user is using for searching.

2 Related Work

One way to support the user when browsing the Web are browsing assistant systems, which suggest links of interest, while the user is browsing. One such system is WebWatcher [26], which suggests a “tour” when browsing a collection of documents by highlighting links. Letizia [14] attempts to anticipate items of interest by exploring links from the user’s current position. This is inferred from browsing behavior. In [6], an automatic approach for rating how interesting a web page is for a specific user is proposed.

When handling a search request, search performance can be improved, if more knowledge about the context of the search is available. Knowledge of the context can either be gained by explicit user input, or learned automatically. The Watson System [25] tries to infer context information from what the user is currently doing on his PC, e.g. documents he is typing or reading. The Persona System [18] uses explicit feedback and an underlying taxonomy to learn the user interests. This information is then used to re-rank search results. In [19], an approach for mapping user interests to categories is proposed. When a user sends a query, the system tries to map it to the right category based on the terms used. This information provides means to disambiguate the query since the category can be understood as the users intention.

Presenting the results structured in categories can also help the user to find information he is seeking faster [17]. This is done, e.g., by the Vivisimo search engine [24], which clusters the results and assigns labels to the found clusters. The clusters are presented as a folder structure. AISEARCH [1,2] also builds categories over search results and then displays it in a graph. Ontologies provide another way to group similar documents together, see, e.g., [30].

Bookmarks of a user give clues about his topics of interest [10]. In [5], using bookmarks in collaborative filtering is studied. In [15], it is tried to learn themes in a community of surfers with overlapping interests by using the bookmarks.

One main problem of almost all approaches mentioned above is that they had been implemented in quite diverse systems and thus an evaluation or comparison of the diverse techniques with respect to usability and retrieval performance in an interactive setting is hardly possible. Especially the effects of the combination of different retrieval support methods cannot be studied due to the systems incompatibilities. A similar integration idea (however in another field of application) is underlying the CANDELA system for video retrieval [4] and the system for media management described in [12].

3 System Architecture

It was our main goal to develop a retrieval system, which is as flexible as possible, to provide a platform for combining different methods for all parts of the multimedia retrieval process. So far, we mainly consider the retrieval of text documents. However, the system is designed such that multimedia data could also be handled. An overview of the system architecture is shown in Figure 1.

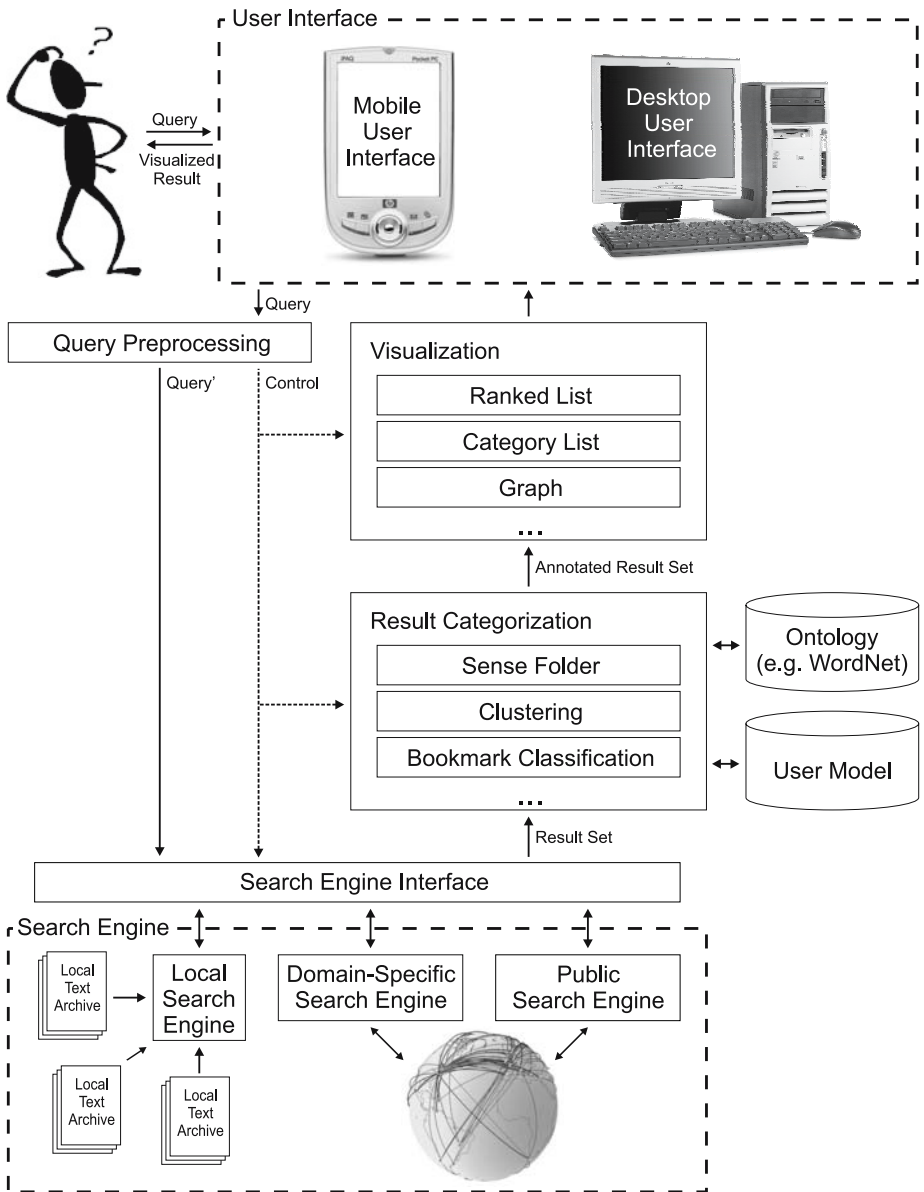


Fig. 1. System Architecture

The system should be accessible from different computers as one user usually accesses the web through different computers, e.g. at home, at the office, or on mobile devices. Therefore, the main application runs on a server. User access can be gained either through a web browser or through a locally installed application (see Sect. 3.1 for details).

The main application handles a pool of plug-ins, each containing one specific method for a specific part of the search process. This comprises plug-ins for accessing document collections for indexing and searching (e.g. local archives, benchmark collections or direct access to search engines like Google), plug-ins that provide methods to structure (e.g. clustering and classification algorithms) or to mine (e.g. methods to extract descriptive concepts) result sets as well as methods that preprocess the data such that it can be easily visualized by a client tool connected to the meta-searcher (e.g. creating graph structures or computing an overview of the result set using self-organizing maps [11]). Furthermore, plug-ins for user profiling (e.g. to access profiles obtained by the proxy based logging mechanism of the CARSA system) and plug-ins that provide access to ontologies (e.g. WordNet) that are required by other plug-ins (e.g. for multilingual support or semantic categorization) can be integrated. Each plug-in can run either on the same server as a dynamically linked java class or on another server accessible through a web service. The first method increases computational efficiency. However, it might not always be possible to run everything on the same server, especially when external resources like the Google API are used. Plug-ins can be registered by a web service or locally with a configuration file.

The plug-ins that should be used for a specific search session can be dynamically selected, i.e., when sending a search request, the user interface can also send information about what plug-ins to use. Otherwise a server defined default setting is used. The search query is forwarded to the selected search engine via the search engine interface (see Sect. 3.2). It returns a ranked list of results. This list, possibly together with an ontology or user model, is used to categorize the results (see Sect. 3.3). After this, the altered list is processed by a visualization module (see Sect. 3.4), which presents the results to the user. Several categorization methods can be called in parallel and will be visualized separately.

All data transfer is done using XML structured data packages. Therefore, the information exchange between different modules can be easily revised or extended. An example of such an XML string, describing the annotated results of a search query, is shown in Fig. 2.

3.1 User Interface

The CARSA architecture enables the development of user interfaces for different contexts (e.g. problem and device specific). These interfaces not only allow the user to specify a search query for a selected search engine, but they can additionally provide the possibility to select and configure the plug-ins registered at the system to suit the user's current needs. Predefined user- and system-related presets help to ease the configuration. These presets are stored on the central server to provide maximum accessibility. In order to give more insight into the ideas and capabilities of CARSA, we describe in the following briefly some interfaces that have been already implemented using this architecture.

First, there is a java servlet based web interface (Fig. 3) that requires no special software to be installed on the client's side. The user can simply access the system with a web browser of his choice and use it like any "ordinary"

```

<?xml version="1.0" encoding="UTF-8"?>
<Result DocumentFiltering="false" EndIndex="10" EstimateIsExact="false"
  EstimatedTotalResultsCount="3870000" SearchComments=""
  SearchTime="0.031655" SearchTips="" StartIndex="0">
  <Query QueryString="universitaet"
    SessionID="7F881092EC34CBA2F421C955A2EC3BDD" UserSessionID="357021445">
    ...
  </Query>
  <RE CachedSize="19k" DirectoryTitle="Freie <b>Universitaet</b> Berlin"
    HostName="" RelatedInformationPresent="true" Snippet="Hauptsite.
    Aktuelle Meldungen, Informationen ueber Studium, Forschung, Links zu<br>
    den Einrichtungen der <b>Universitaet</b>, Online-Vorlesungsverzeichnis
    sowie <b>...</b>" Summary="Hauptsite. Aktuelle Meldungen, Informationen
    ueber Studium, Forschung, Links zu<br> den Einrichtungen der
    <b>Universitaet</b>, Online-Vorlesungsverzeichnis sowie Hinweise zu
    <b>...</b>" Title="Freie <b>Universitaet</b> Berlin"
    URL="http://www.fu-berlin.de/">
  <PluginResultList>
    <PluginResultElement
      JavaClassName="de.unimd.irgroup.carsa.classification.SetClassification"
      Plugin="BayesClassifier">
      <CE ClassName="Uni" Probability="1.0"/>
      <CE ClassName="Berlin" Probability="0.999999999984375"/>
    </PluginResultElement>
    <PluginResultElement
      JavaClassName="de.unimd.irgroup.carsa.classification.SetClassification"
      Plugin="kNN">
      <CE ClassName="Berlin" Probability="0.84"/>
      <CE ClassName="Magdeburg" Probability="0.16"/>
    </PluginResultElement>
  </PluginResultList>
</RE>
  ...
</Result>

```

Fig. 2. Sample XML-string describing annotated results

internet search engine. A login provided at the web site enables access to a stored user profile. Furthermore, the interface provides transparent access to select and configure the installed plug-ins (see Fig. 4). This user interface can be used as both, mobile and desktop user interface.

Furthermore, a client based interface that contains means to manage bookmarks stored on the server as well as providing access to our search plug-ins (Fig. 5) was implemented. Results of a search request are shown in the web browser using the java servlet (Fig. 3) described above. However, based on the bookmark management, this tool enables the use of bookmark based classification methods (see also Sect. 3.3).

The mobile user interface shown in Figures 6 – 8 has been especially designed to meet the requirements that go along with the limited display size of mobile devices such as PDAs and the specific type of interaction using a pen [20]. It has

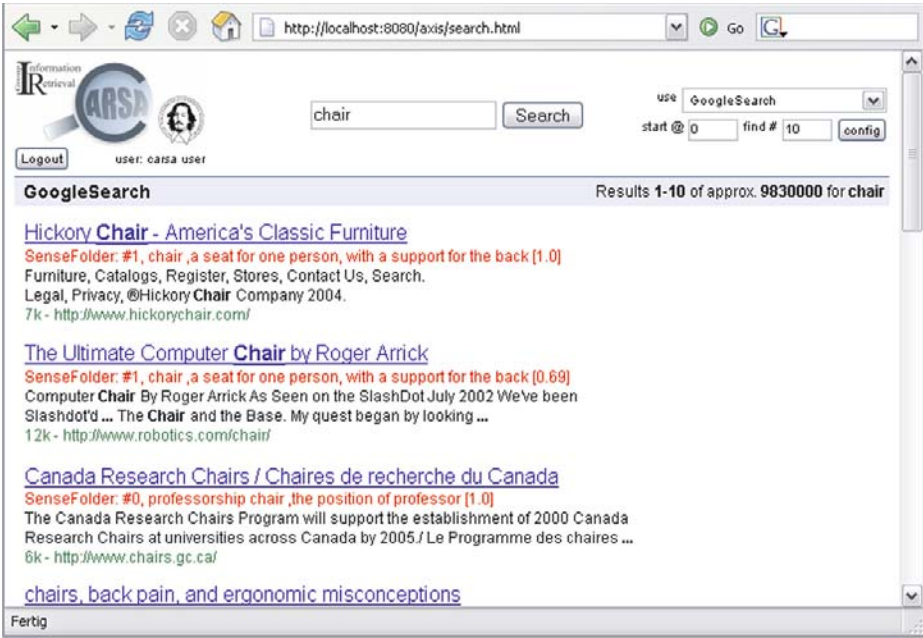


Fig. 3. Web Based Interface: Searching

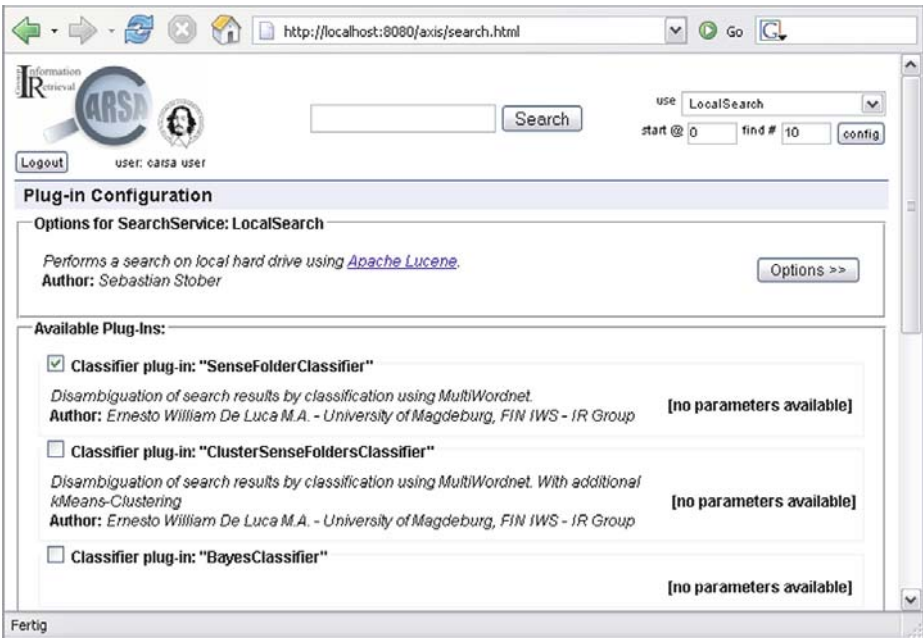


Fig. 4. Web Based Interface: Configuration Dialog

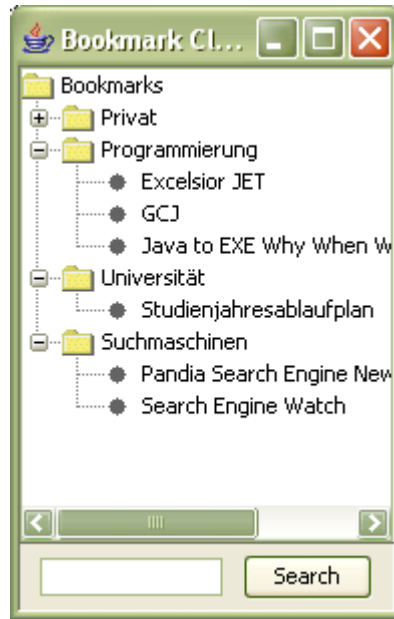


Fig. 5. Bookmark Client

been implemented in Macromedia Flash. Hence, in addition to the application, the Macromedia Flash Player has to be installed on the user's mobile device. The user interface is divided into three views. In the first view (Fig. 6), the user can specify his query and select a page of search results that then is displayed in the second view (Fig. 7). Results may be added to bookmark categories that are displayed in the third view (Fig. 8).

There is also a desktop version of the mobile user interface available that puts the three views of the mobile user interface together into a single frameset. It also requires installation of a Macromedia Flash Player.

All user interfaces presented here still provide a ranked list of search results. Further information gained through one or several categorization methods are simply added as additional information to the document snippets. However, we are currently working on other visualization techniques, e.g. for visualizing classification of search results into an existing class structure.

3.2 Search Engine Interface

The search engine interface allows searching for data over different search engines. Each possible search engine connection is handled by plug-ins. Each plug-in modifies the input to fit the specific search engine and ensures that the output of the search engine is converted to the CARSA specific internal format.

Furthermore, for web based searching the CARSA system automatically downloads, indexes, and caches the web pages of the search hits and provides these information to the plug-ins for further processing, e.g., for classification or



Fig. 6. PDA Client - Search



Fig. 7. PDA Client - Results

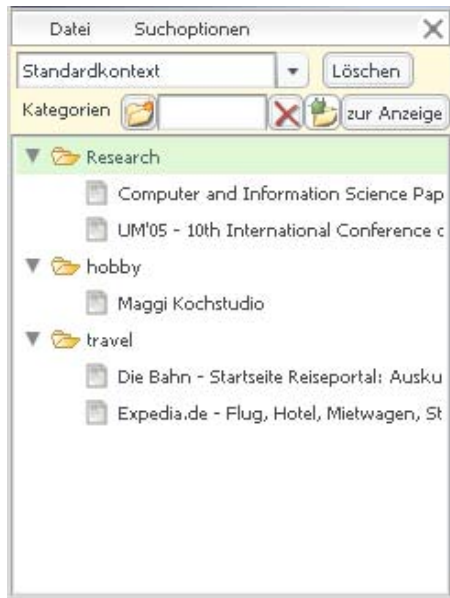


Fig. 8. PDA Client - Bookmarks

re-ranking. Due to the multithreaded architecture and the cache this approach is reasonably efficient also for online searching.

The following search plug-ins have been implemented so far:

- To search the whole web, we currently use the Google API [8].
- Furthermore, we have implemented our own webcrawler, which runs on a cluster PC. It is designed to crawl a domain-specific subset of the web. The domain is described by linguistic quantifiers.
- We also have developed a local searcher working on a stored document collection. On the one hand, this enables testing on a defined data set (benchmark collection). On the other hand, the local search can be used to search data distributed on several computers in the Intranet.

3.3 Plug-Ins for Categorization, Clustering and Re-ranking

In order to allow a simple post-processing of result sets, plug-ins for categorization, clustering and re-ranking can be easily installed. These plug-ins have full access to user specific information (e.g., bookmarks and user profiles) and to further information resources like ontologies. The plug-ins receive a fully indexed set of the retrieved documents and can compute annotations (e.g. class or cluster information) or re-rank the set. In the following, we briefly describe some annotation methods that have already been implemented.

Categorizing search results can be done in several ways. Without using any additional information, one can find clusters of similar documents based on their $tf \times idf$ -vectors [3]. The found clusters can then be labeled, e.g. by the method proposed in [13].

Another approach studied by us is based on the use of semantic information provided by an ontology. As language itself is highly ambiguous so are the search terms. However, a user has usually only one meaning in mind. Therefore, categorizing the search results in groups of meaning (the so called *Sense Folders*) can help extracting the possibly interesting documents. The different meanings of a word are stored in an ontology. In our case, we use the WordNet ontology [29, 28], which is a general ontology. More details about the sense folder approach can be found in [16, 9].

A third approach studied by us is based on a user profile, which is constantly built while the user is browsing the web. Part of this profile are the bookmarks. If structured, they provide some hints about topics of interest of the user and how he distinguishes one from another. We currently use an approach based on Naïve Bayes classifiers to assign the bookmark categories provided by a user to the entries in a result set [21].

3.4 Visualization

The easiest way of visualizing the results is a ranked list of the result documents. Information gained from categorization can be displayed in an additional row for every document stating the assigned class. However, this is just slightly helpful as the user still has to scan all results.

Displaying a category tree with the documents on the lowest level can speed up the search process [17], because the user only needs to look at selected results as the category tree gives some clues about the content of the specific web sites.

Bringing this category tree a step further is to display them in form of a graph. Here, more information can be visualized in a single figure, e.g. similarity of topics can be shown by mapping similar topics closer together than distinct topics (see tools like SPIRE [23], SCI-Map [22], VxInsight [7] and mappings using self organizing maps [11, 27]).

4 Conclusion

In this paper, we have given a brief overview of the CARSA system, an architecture that supports the development and evaluation of context adaptive information retrieval systems. We have motivated and described the underlying architecture that allows the integration of diverse methods for user support in information retrieval tasks in a unified framework. In order to show its usability in practice, we briefly described some interfaces and annotation methods that we have already implemented based on this architecture. However, more detailed user studies and large-scale performance experiments on the classifiers still have to be done.

Further information about CARSA is available on the web pages of the information retrieval group Magdeburg: <http://irgroup.cs.uni-magdeburg.de/>.

References

1. B. Stein, S. Meyer zu Eissen: AISEARCH: Category Formation of Web Search Results, 2003.
2. AISEARCH, <http://www-ai.upb.de/aisearch/>.
3. G. Salton, A. Wong, C.S. Yang: A vector space model for automatic indexing, In: Communications of the ACM 18(11), 1971.
4. E. Jaspers, R. Wijnhoven, R. Albers, J. Nesvadba, J. Lukkien, A. Sinitsyn, X. Desurmont, P. Pietarila, J. Palo, and R. Truyen: CANDELA - Storage, Analysis and Retrieval of Video Content in Distributed Systems In: Proc. of the 3rd International Workshop on Adaptive Multimedia Retrieval (AMR), 2005.
5. J.J. Jung, J.-S. Yoon, G. Jo: Collaborative Information Filtering by Using Categorized Bookmarks on the Web, In: 14th International Conference of Applications of Prolog (INAP), 2001.
6. P.K. Chan: Constructing Web User Profiles: A Non-invasive Learning Approach, In: Proc. of the International Workshop on Web Usage Analysis and User Profiling, 1999.
7. K. W. Boyack, B. N. Wylie, G. S. Davidson: Domain Visualization Using VxInsight for Science and Technology Management, Journal of the American Society for Information Science and Technologie 53(9):764-774, 2002.
8. Google API, <http://www.google.com/apis/>.
9. E.W. De Luca, A. Nürnberger: Improving Ontology-Based Sense Folder Classification of Document Collections with Clustering Methods, In: Proc. of the 2nd Intern. Workshop on Adaptive Multimedia Retrieval (AMR), 2004.
10. D. Abrams, R. Baecker, M. Chignell: Information Archiving with Bookmarks: Personal Web Space Construction and Organization, In: Proc. of the SIGCHI conference on Human factors in computing systems, 1998.

11. A. Nürnberger: Interactive Text Retrieval Supported by Growing Self-Organizing Maps, In: Proc. of the International Workshop on Information Retrieval, 2001.
12. W. Fontijn, J. Nesvadba, A. Sinitsyn: Integrating Media Management Towards Ambient Intelligence; In: Proc. of the 3rd International Workshop on Adaptive Multimedia Retrieval (AMR), 2005.
13. K. Lagus, S. Kaski: Keyword Selection Method for Characterizing Text Document Maps, In: Proc. of the ninth international conference on artificial neural networks (ICANN'99), Vol. 1, 1999.
14. H. Lieberman: Letizia: An Agent That Assists Web Browsing, In: Proc. of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.
15. S. Chakrabarti, Y. Batterrywala: Mining themes from bookmarks, In: Proc. of the Workshop on Text Mining held at the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2000.
16. E.W. De Luca, A. Nürnberger: Ontology-Based Semantic Online Classification of Documents: Supporting Users in Searching the Web, In: Proc. of the European Symposium on Intelligent Technologies (EUNITE), 2004.
17. S. Dumais, E. Cutrell, H. Chen: Optimizing search by showing results in context, In: Proc. of the SIGCHI conference on Human factors in computing systems, 2001.
18. F. Tanudjaja, L. Mui: Persona: A Contextualized and Personalized Web Search, In: Proc. of the 35th Hawaii International Conference on System Sciences, 2002.
19. F. Lui, C. Yu, W. Meng: Personalized Web Search by Mapping User Queries to Categories, In: Proc. of the 11th International Conference on Information and Knowledge Management, 2002.
20. E.W. De Luca, A. Nürnberger: Supporting Information Retrieval on Mobile Devices In: Proc. of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services, 2005.
21. K. Bade, A. Nürnberger: Supporting Web Search by User Specific Document Categorization: Intelligent Bookmarks In: Proc. of the Leipziger Informatik Tage (LIT), 2005.
22. H. Small: Visualizing science by citation mapping, *Journal of the American Society for Information Science* 50(9):799-813, 1999.
23. J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur and V. Crow: Visualizing the non-visual: Spatial analysis and interaction with information from text documents, In: Proc. of IEEE Symposium on Information Visualization, 1995.
24. Vivisimo, <http://vivisimo.com>.
25. J. Budzik, K. Hammond: Watson: Anticipating and Contextualizing Information Needs, In: 62nd Annual Meeting of the American Society for Information Science, 1999.
26. T. Joachims, D. Freitag, T. Mitchell: WebWatcher: A Tour Guide for the World Wide Web, In: Proc. of the 15th International Joint Conference on Artificial Intelligence, 1997.
27. A. Nürnberger and M. Detyniecki: Weighted Self-Organizing Maps: Incorporating User Feedback, In: Proc. of 13th Int. Conf. on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP 2003), pp. 883-890, Springer-Verlag, 2003.
28. WordNet homepage, <http://wordnet.princeton.edu/>.
29. C. Fellbaum: WordNet, an electronic lexical database, 1998.
30. Y. Labrou, T. Finin: Yahoo! as an ontology: using Yahoo! categories to describe documents, In: Proc. of the 8th Int. Conf. on Information and Knowledge Management, 1999.

Integrating Media Management Towards Ambient Intelligence

Willem Fontijn, Jan Nesvadba, and Alexander Sinitsyn

Philips Research, Prof. Holstlaan 4 (WDC 1),
5656 AA Eindhoven, The Netherlands
{willem.fontijn, jan.nesvadba,
alexander.sinitsyn}@philips.com

Abstract. As Consumer Electronics devices get interconnected, the issue of aligning their data management solutions becomes prominent. If we want to deploy comprehensive applications that are perceived to be intelligent we need to integrate data management. Because this has to be done across multiple platforms and encompasses both legacy and future devices, we cannot get away with hard-coded data management functionality in each individual application. We need to add high-level data management functionality to the distributed middleware layer. Our peer-to-peer database management system, called AmbientDB, addresses this necessity by providing a global database abstraction layer over an ad-hoc network of heterogeneous peers. Such peers range from stationary media servers, via mobile AV jukeboxes to sensor networks. We will present the first steps of this integration process applied to real product types.

1 Introduction

Future Consumer Electronics (CE) environments are expected to be characterized by the omnipresence of processing power, connectivity and storage. However, while these devices get more complex, the requirement of ease of use gets more stringent. This is the basic premise of Ambient Intelligence (AmI), which is the focal point of various ongoing research [1].

AmI has two aspects. First, the technology needs to be part of the ambience, i.e. unobtrusive, both physically, and in its relation to the user. Second, we want to perceive the system to be intelligent in its interaction with the user. To create the perception of intelligence the AmI system has to present to the user a unified and coherent view irrespective of the context and location of the user or the type of interaction. To achieve this we need to integrate and relate data from a wide variety of sources, ranging from simple sensor nodes to multimedia servers. To meet this challenge AmbientDB, peer-to-peer (P2P) data management middleware, is being developed [2].

The Connected Planet (CP) [3] may be considered a first step towards AmI. By providing wireless links between the PC, TV and audio systems, CP makes digital multimedia in every form and from any source, including broadband Internet, available throughout the entire home (see Figure 1a). At this stage we observe the introduction of embedded databases in CE devices. While CP provides connectivity between these databases at a physical level, for AmI we will eventually need to

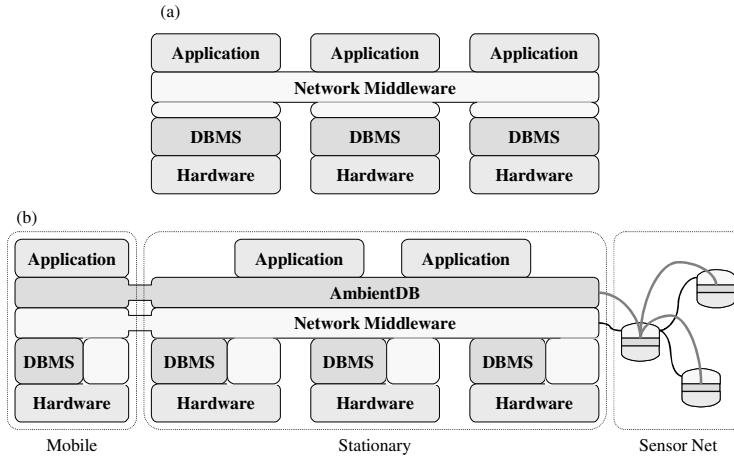


Fig. 1. (a) Representation of Connected Planet concept. Applications run on their own hardware, with an embedded database and physically interconnected through a network layer. (b) Representation of the concept of AmbientDB. Some applications run on dedicated hardware like in the case of mobile devices. Some applications run in a distributed fashion. All data sources, including sensors, are logically interlinked, presenting an integrated view on all data throughout the AmbientDB data management layer.

connect and to integrate the constituent embedded databases at a logical level (see Figure 1b). Already before that we need improvements in embedded databases to cope with the metadata for the huge amounts of content from various sources and to support the adding of more structure and meaning to metadata, required to support new applications.

Currently we witness an explosive growth of the amount of each of the various types of data. First, the number and size of items of content, second, the amount of metadata to support the identification, classification, navigation and retrieval of content, and third, the amount of context data from sensors and usage tracking to support customization, personalization and adaptive behavior, all steadily and rapidly increase.

An example of a new source of metadata is the output of content analysis. Terabytes of storage capacity, millions of Million Instructions Per Second (MIPS) of processing power and gigabits of bandwidth become a reality in the CP [3]. The resulting explosive growth of digital multimedia assets, creates the need for advanced methods of browsing, navigating, retrieval, and finally for content management. Essential to enable those advanced methods are content descriptors, which constitute metadata that provide additional information about audiovisual signals and content. These content descriptors can not be created by hand, due to the amount of content and required detail of description. Instead they need to be generated by Video or Multimedia Content Analysis (VCA respectively MCA). Once generated, low-level audio/visual content descriptors are often stored and later reused in high-level semantic content analysis. This adds real-time performance requirements on metadata management. Furthermore, the usage of AmI-based distributed-MCA technologies on

a network of (legacy) CE devices and sensors by means of smart usage of scattered processing power across In-Home networks [4] provide e.g. Personal Video Recorders (PVRs), portable/mobile devices and other CE devices with capabilities and functionalities not feasible if done in isolation. Such distributed/mobile MCA applications require high availability of data and metadata storage and retrieval. The combination of high performance and universal availability is one of the unique requirements on media management towards ambient intelligence.

This paper gives an integrated overview of research being done in half a dozen product related projects, which are connected through the database technology used. Thus, it highlights the challenges of the next step in media management, which will function as a stepping-stone for the larger, more distant challenge of AmbientDB. We will describe the integration process of various applications across various platforms into one data management framework. In section 2 we introduce the various platforms and applications we incorporate in this development, as well as the main challenges they contribute. Section 3 deals with the solution direction we have selected to integrate these platforms and applications and we exemplify the advanced cross device services our solution facilitates.

2 Problem Description

Traditionally in CE a separate data management solution was developed for each platform and each application. For AmI we need to align and integrate these solutions at a logical level. However, before we can integrate them we need to analyze their specific requirements.

2.1 Platforms and Applications

We consider a wide range of devices exemplified by three specific categories: stationary, mobile and distributed. For each of these categories we have projects running and from these we take an existing platform with a key application as a representative for the category.

2.1.1 Stationary - PVR

Our representative in the stationary device category is the entertainment hub (eHub). This can be seen as the convergent device of PVR, set-top box, and game console. It has a central role in the entertainment of consumers while these are at home, e.g. sitting on the couch. It is a relatively resource rich device connected continuously to (broadband) Internet and to the power mains. A Bluray Disc / Hard disk combination device falls into this category.

A common use of the eHub will be that it is the central source of audiovisual data for entertainment purposes. It will autonomously record TV programs for its users and catalog and arrange video sources in general to make these available at a button press, throughout the home. But also other media types, like audio and still pictures, will be stored and managed by this device. Alongside the PC it will be a main content repository in the home. Currently, we are in the process of aligning the eHub data management with that of the PiC.

2.1.2 Mobile - AV Jukebox

We use the Personal infotainment Companion (PiC) as the representative in the mobile device category. It is a relatively small, personal device that stores and renders digital media (audio, video, still pictures and other data) and is used both at home and away. The device contains a large-capacity embedded storage facility and has advanced (wireless and wired) networking capabilities. The PiC project is the forerunner in data management integration.

The market for "mobile infotainment products" is developing rapidly. The emerging devices provide an entirely new category of mobile audio-visual (AV) entertainment experiences. In general one could characterize them as AV jukeboxes. They strive to provide the user with the experience of having all media that are needed available at any time, in any place, regardless of connection availability in the heterogeneous environment. One of the main challenges of these mobile infotainment devices is to keep data exchange and synchronization transparent to the user.

2.1.3 Distributed - Sensor Networks

Sensor nodes represent the category of most resource constraint devices. These nodes are constraint on energy, processing and storage. Their main trait is that they are all (inter)connected and their main task is to collect context information, which they use themselves or make available to other devices. Though at this stage, our projects on sensor nodes are still disconnected from the other projects, we are in the process of making the connection.

There are many conceivable uses of sensor networks. One would be to collect context information to enable preference-based selection of multimedia content. A sensor network deployed in the home can answer questions like: who is at home, what is his/her mood, what is the light level. Based on what a specific user likes in specific circumstances a detailed profile can be created. Once created the profile is matched with live context information to optimize content selection.

Another use is in lighting control. We can expect widespread use of wireless sensor networks in professional lighting applications but also in the home. Solid-state lighting is attractive because it allows for a wide choice of color, intensity and direction of the light. It is however not obvious how the user should cope with so much choice. We can think of the use of wireless sensor networks to limit the choice based on sensed context but also as tangible user interfaces to actually make the selection. An appealing option that also becomes available is the combination of lighting control and content rendering. An example is Ambilight TV where the perceived screen size increases because light sources surrounding the screen reflect the color scheme on the screen.

2.2 Issues

The device types and applications above give of the range of issues we want to resolve in an integrated manner. We will describe the most important ones here. The integration of data management for the PiC and the eHub is already on its way so we present their issues jointly. The distributed case of sensor networks clearly adds a different set of requirements.

2.2.1 Stationary - Mobile

One main issue for stationary devices is coping with the sheer number of content items, or assets, stored: how to manage and navigate large collections of assets, finding what you need quickly and conveniently. But as the capacity of the mobile devices increases rapidly, the same issue emerges for that platform, with the added problem of having a limited screen size for user interaction.

In combination this raises the issue of synchronization: how to keep your stationary and mobile collections aligned [12]. In essence, to the problem of volume, the problem of distribution is added. And, if you synchronize, how do you do it. It needs to be robust, multi-target, priority-based and resource-aware but also quick, transparent and configurable by the user.

There is also the issue of interoperability: how to cope with connectivity via multiple communication protocols with various types of devices of different sizes from different vendors? How to cope with the exchange of data in different formats (e.g. DIDL, MPEG7/21, MPV [5])?

Apart from the format of the metadata, we need to know what metadata we actually require, more than just the content needs to be described. For instance, we need to manage also, information about people (e.g. users, contacts, artists), and all in an extendable way (extensible types, attributes, schemas). Furthermore, we cannot suffice with a per application metadata set. We require a comprehensive schema or ways to merge schemas.

Large collections also require automated methods to keep the collection clean: how to deal with noisy data (e.g. artists 'Madonna' and 'madona' are the same).

2.2.2 Distributed

Looking at sensor networks we get a whole new set of requirements. This is the main reason why we expect that the integration of sensor networks in the scheme of AmbientDB will take the most time. However, we want to take these requirements into account already now to ensure future compatibility.

A major issue in the sensor network community at the moment is power consumption. Typical applications considered are geographically spread and raw sensor data is sent to a single root node, resulting in considerable cost of sending data over the wireless connections. However, for our case of indoor applications we may assume that some nodes that are connected to a wired power supply are in communications range at all times. This can alleviate the issue of lifetime with battery powered wireless nodes. With a high density of wireless nodes the main problem then becomes bandwidth. The challenge becomes to reduce the amount of data communicated.

To reduce the bandwidth consumption we should interpret the raw sensor data as close to the source as possible. These interpretations need to be aligned with the stationary and mobile devices to enable those to use them. At the same time we want to accommodate applications that require direct access to raw sensor data.

Regarding sensor networks, we need the ability to integrate new sensors and new sensed data types. We need to support data mining. And, for the ability to upgrade our interpretations, we need a mechanism for remote deployment of code.

Finally, to be sure that our data management solution fits each platform well, it needs to be scalable (from mobile audio player to home server) and able to incorporate the extreme cases of sensor nodes.

3 Architecture and Directions

We have seen that database technology is permeating the whole range of device categories. The most effective and efficient way to align the data management at a logical level is to develop a single database technology to cover all future devices. To achieve this we have to integrate the requirements into one superset.

The first step in this process has been to standardize on one (existing) database technology for the prototypes from selected projects in the mobile and stationary category. The next step, currently underway, is to standardize on one schema encompassing all sample applications and then to develop an alternative technology (x100) [6] that will provide us with a framework to develop the added functionality we require. At that stage we expect to be able to include the sensor networks into the picture. The final step is to move all remaining projects and future products to our framework.

3.1 Media Distribution Architecture

After analyzing the user scenarios and requirements, we made a number of decisions, which enabled us to design an architecture for the media management.

One of the challenges is to support the user in handling large sets of digital assets via user-initiated actions and via automatic strategy-based transfers [7]. In order to offer the user yet bigger freedom to manage his digital collections we need to support both on-line and off-line digital asset exchange and synchronization by providing users with views on non-connected devices/collections. The storage requirements for metadata are an order of magnitude smaller than that size of content, which makes it feasible to have higher information availability and offers more freedom in building dynamic views. Therefore, we decided to separate the distribution of content and of metadata. This provides more possibilities for the user to make off-line decisions while managing his collections.

For scalable and robust information management across multiple heterogeneous data sources our middleware needs to abstract from the underlying storage technologies (e.g. file system, Database Management System (DBMS)).

The device (dis)connection process should be transparent to the user. The user should be able to manage assets, even if these are currently unavailable due to disconnection. To support off-line asset exchange our system uses a metadata snapshot mechanism, which provides recent views on all content. The user can perform actions on assets of a disconnected device and the system propagates these actions when the device connects. Due to the low cost of storage it is worthwhile to have a snapshot for each known device, on each device. Snapshots have yet another advantage: even if the queried device is accessible, the access time to metadata stored in a local snapshot is shorter compared to accessing the remote device via the network. Of course, snapshots should be synchronized regularly.

An architecture for the distribution of media based on the above design decisions has been defined, presented in [8] and is being implemented. One of the essential components of this system is the Metadata Store, which deals with the storing of metadata to make it available to local or remote applications via high-level APIs. This

component includes a general purpose DBMS with a schema-dependent metadata abstraction layer on top of it. This abstraction layer provides access to the local database and snapshots via an extendable set of object-oriented interfaces. Furthermore, the abstraction makes it possible to use any DBMS underneath, without the need to change the interface (see Figure 2).

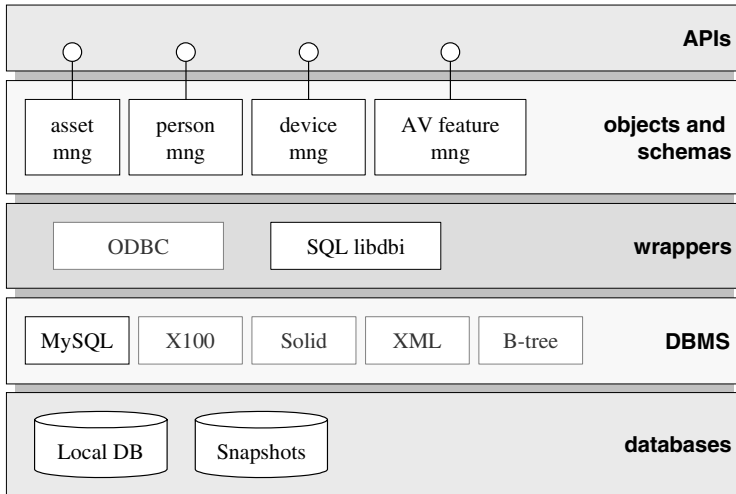


Fig. 2. Metadata Store Framework. Grey outlined components are not yet implemented.

3.2 DBMS-Based Cross-Domain Applications and Services

Various mid- and high-level multimedia VCA/MCA solutions have been researched in a variety of national respectively international collaborations such as MultimediaN [9] and Candela [5]. These solutions will provide consumers with personalized, responsive and anticipatory applications. An example of such an application is face-detection-enabled sensors that initiate another device to remotely or autonomously identify the person by means of biometrics [10]. Home-environment-parameters are then set to the personal, context dependent, preferences of the user detected. To capture the context sensor networks are used.

To enable such applications our Metadata Store has been adapted. To store streamed MCA data a cache mechanism was added and an AV Feature management interface has been implemented (see Figure 2). The data model has been extended to support mid- and high-level AV features in different formats (e.g. proprietary, DIDL and MPEG7). Furthermore, personal data are synchronized automatically between portable/mobile devices (e.g. PiC) and various stationary devices (e.g. eHub / PVR) in the home. Personalized high-level MCA routines are initiated transparently to enable meaningful intuitive search queries such as a personalized music or AV clip playlist generation of content stored on the PiC, analyzed on the eHub and rendered on the PiC again.

3.3 Directions

The architecture described above can resolve many of the issues listed in paragraph 2.2. Some of them have been addressed in [12]. In this section, we will discuss several important issues that are relevant to media management towards ambient intelligence.

3.3.1 Volume

One way to cope with the sheer number of content items with regard to management and navigation is to take context into account. Context can be derived from, for instance, the capabilities of the current device, profiles, preferences and all kinds of sensor data. One end of the spectrum is fully automatic selection of media based on the combination of, for instance, identity of the user, location, time, mood, and so on. But already offering a reduced view based on one context parameter is useful.

Another way is to use content analysis to attach standardized labels to items of content, which can be used to create many different intuitive cross sections based on content properties. Also non-standard labels can be used. Making use of sensors and profiling one can create cross sections based on the users preferences regarding certain items of content.

3.3.2 Consistency

How to insure consistency of snapshots is an important issue. First though, we have to realize that full consistency across all devices is not a requirement. We just need to ensure that the local view the user is confronted with is consistent. For instance, the user may use different naming conventions depending on the access method or location. The local database on the PiC may describe the same content differently then the snapshot of the eHub database does. How this is presented to the user is a matter of choice. Either the local description has precedence or the user is made aware of the distinction between the local database and the snapshot. To facilitate this, we need unique content identifiers in order to link identical items of content unambiguously, irrespective of the description. In the case of music, audio fingerprinting is used for this, which can identify music irrespective of source or quality. For other media other content analysis methods can be used. Unambiguous content identification can also be instrumental to resolve the issue of noisy data, e.g. incomplete, conflicting and wrong descriptions.

We will discuss the snapshot consistency issue somewhat further using an example. Take for instance a music collection. The mobile PiC contains a sub set of the music owned by the user. The stationary eHub contains all music. On the PiC there is a local database describing the local content and a snapshot of all content on the eHub. As stated above the descriptions of the same content may differ between the two. If the description of a song on the PiC is changed, the local database will be updated. All snapshots of the database on the PiC are now inconsistent with the database itself. Whether and how this update is propagated is decided by the update strategy. In the mean time the database on the eHub may have been updated. However, until the PiC reconnects to the eHub we can work under the assumption that the snapshot is correct. As the local data is only used for accessing the local content any inconsistency between a local snapshot and the remote original are immaterial until the devices reconnect. There are possible conflicts if both the eHub database and

the local snapshot of the eHub database on the PiC are updated. One option to resolve this is to use time stamped logging of the updates per user. For a single user the latest update will prevail. For multiple users we can store both versions and which is presented depends on the context, i.e. which user is active. This way the view of each user will remain consistent with the perception of the user.

3.3.3 Interoperability

As physical interoperability is a separate issue that lies outside of the scope of this paper we will focus on interoperability on the level of data. The translation of data in one metadata format into another can be easily accommodated by creating detailed mappings from one format to another. The automatic generation of mapping tables is also possible if enough data with the same descriptors in the different formats is provided. However, given the limited set of widespread metadata formats it is questionable whether the creation of such an automatic mapper is worthwhile.

A bigger problem is how to decide on which metadata is required and how to ensure we can extend these requirements over time in devices that are already deployed. Based on the combination of applications we envisage, we are working on a comprehensive schema describing all data used and anticipated currently, which is sufficient for now and should be sufficient for the near future also.

3.3.4 Sensors

If we assume the all sensors are one hop away from a less resource-restricted node, the power consumption of the most restricted devices, the sensor nodes, is less of an issue. Then limited bandwidth becomes the biggest issue. By moving sensor data interpretation as close to the data source as possible and pushing database functionality into the sensor network to be able to apply query optimization methods in the network, we can effectively reduce the dataflow [11]. The use of a feature rich database middleware layer on top of the sensor network facilitates also the transparent deployment of so called history nodes, which log sensor data and can cache query results. Access to the sensor nodes themselves can thus be limited to the minimum while preserving the full functionality of the sensor network.

4 Conclusions

Recent and expected trends in the CE domain require new database solutions. The isolated databases currently used will be merged into clusters that will act and response as one single virtual database to the network. This transparent distributed data management is crucial to AmI applications. In the end we will connect all (heterogeneous) data sources in the AmI environment to present a single, integrated view on all data in the environment, enabling applications to take full advantage of all information available with minimal effort. We have taken the first steps in integrating the data management for our full range of platforms. It has become clear that this is not only efficient but also necessary. Only thus we can deploy new, cross platform, ambient applications and services that create a perception of intelligence. Required middleware, interfaces, protocols, DBMS cores and data models are only some examples of issues that require further research to make this vision come true.

References

1. Philips Ambient Intelligence vision, www.philips.com/research/ami
2. W.F.J. Fontijn, P.A. Boncz. AmbientDB: P2P Data Management Middleware for Ambient Intelligence. Middleware Support for Pervasive Computing Workshop at the 2nd Conference on Pervasive Computing, (PERWARE04), pages 203-207, Orlando, USA, March 2004.
3. Philips Connected Planet vision, www.philips.com/connectedplanet
4. F de Lange, J. Nesvadba. A Hardware/Software Framework for the Rapid Prototyping of Multimedia Analysis Systems. Workshop of Image Analysis for Multimedia Interactive Systems (WIAMIS), Montreux, Switzerland, April 2005, submitted.
5. CANDELA project, www.extra.research.philips.com/euprojects/candela
6. P. Boncz, C. Treijtel. AmbientDB: relational query processing in a P2P network. Proc. DBISP2P Workshop 2003, Berlin (co-located with VLDB'03), LNCS 2788, Springer Verlag 153 - 168.
7. A. Sinitsyn. A Synchronization Framework for Personal Mobile Servers. Middleware Support for Pervasive Computing Workshop at the 2nd Conference on Pervasive Computing (PERWARE04), pages 208-212, Orlando, USA, March 2004.
8. A. Sinitsyn, W. Berkvens, A. Claassen, J. van Gassel. Media Distribution in a Pervasive Computing Environment. Middleware Support for Pervasive Computing Workshop at the 3rd Conference on Pervasive Computing (PERWARE05), pages 204-208, Kauai Island, Hawaii, March 2005.
9. MultimediaN project, www.multimedien.nl
10. J. Nesvadba, P. Miguel Fonseca, R. Kleihorst, H. Broers, J. Fan, Face Related Features in Consumer Electronic (CE) device environments, Proc. IEEE Int'l Conf. on Systems, Man, and Cybernetics, pp 641-648, The Hague, Netherlands, 2004.
11. N. Chen, W.F.J. Fontijn, Q. Zhang, and X Chen. A Framework for Ambient Applications, International Conference on Sensor Networks (SENET05), Montreal, Canada. August 2005, submitted.
12. G. Bernard et al. Mobile databases: a selection of open issues and research directions. SIGMOD Rec. 33, 2 (Jun. 2004).

CANDELA - Storage, Analysis and Retrieval of Video Content in Distributed Systems

E. Jaspers¹, R. Wijnhoven¹, R. Albers^{1,1},
J. Nesvadba², J. Lukkien³, A. Sinitsyn²,
X. Desurmont⁴, P. Pietarila⁵, J. Palo⁶, and R. Truyen⁷

¹ Bosch Security Systems, Eindhoven, The Netherlands

² Philips Research, Eindhoven, The Netherlands

³ Eindhoven Technical University, Eindhoven, The Netherlands

⁴ Multitel, Mons, Belgium

⁵ VTT, Oulu, Finland

⁶ Solid, Oulu, Finland

⁷ Philips Medical Systems, Eindhoven, The Netherlands

Abstract. Although many different types of technologies for information systems have evolved over the last decades (such as databases, video systems, the Internet and mobile telecommunication), the integration of these technologies is just in its infancy and has the potential to introduce "intelligent" systems. The CANDELA project, which is part of the European ITEA program, focuses on the integration of video content analysis in combination with networked delivery and storage technologies. To unleash the full potential of such integration, adaptive video-content analysis and retrieval techniques are being explored by developing several pilot applications.

1 Introduction

After the introduction of Digital Video Broadcasting, video enhancement and interactive video enabled the user to interact with the video delivery system. As a next step, there exist a growing desire for content retrieval, applying search queries that are natural for humans. Because the type of queries that are natural for humans depend on the application domain, the analysis and retrieval functionality should be adaptive to the application. For example, in a home video application one might search for a specific genre, whereas in a surveillance application suspicious behavior is searched. This requires understanding of the content and implies video content analysis (VCA) techniques like segmentation into video objects, metadata generation for large databases of video content and the use of search and presentation devices.

Currently, the development of digital video analysis is mainly focused on state-of-the-art video compression (MPEG-4/H.264), describing the video

content (MPEG-7), and standardizing a framework to enable interoperability (MPEG-21). All these standards are very much related to the scope of CANDELA, but do not address analysis algorithms or the adaptivity to the application domain. Even though several VCA algorithms have been proposed and a standard for describing the content is available, complete system solutions remain proprietary. For example, how can we detect a pulmonary embolism in the huge amount of pictorial data from a medical CT scanner? How can we detect and identify a shoplifter in a warehouse without manually observing hundreds of security cameras? How can we retrieve information about our favorite holiday place on a mobile device by applying abstract search queries on huge databases?

In addition to these application-specific algorithmic and standardization issues, the architecture of these VCA systems is of importance. First, a VCA system will consist of several components both logically and physically. Video acquisition, coding and storage, computing the mentioned metadata and visualizing the combination of data and metadata are among the functionalities that can be recognized. It is to be expected that these functionalities are partitioned across networked components for reasons of geographical distribution, cost and performance. Coding and displaying have to be adapted to the capabilities of the system at hand. Even detailed functionalities like complicated VCA algorithms can be distributed across low-cost components leading to distributed processing. This is why the networked delivery and the distribution is an integral part of CANDELA architectures.

In this paper, we will describe a complete retrieval system, but focus on its adaptive aspects. Firstly, Section 2 will describe the system architecture with particular emphasis on the impact of the distributed aspects and corresponding adaptivity, e.g. in the resource allocation. Secondly, the paper will elaborate on the application-specific content analysis in Section 3. Thirdly, Section 4 shows how the retrieval application differs per application domain. One of the subsections explains how ontologies can be exploited to introduce human-like reasoning into the system, i.e. provide the relation between human-natural search queries and the content descriptors. Section 5 will finalize with some conclusions.

2 System Architecture and Adaptivity

Fig. 1a shows the general system architecture, which was already identified by Petrovic and Jonker [1], comprising the integration of content analysis, storage, querying and searching. In Fig. 1b the CANDELA system architecture is depicted which is largely in accordance with Fig. 1a. The figure indicates both the information streams through the system as well as the relevant system components. The system is highly distributed, even more as some of the components themselves have a distributed realization. The distribution brings several concerns about the functionality, which are addressed within the architecture as well. These are described in more detail in subsequent sections.

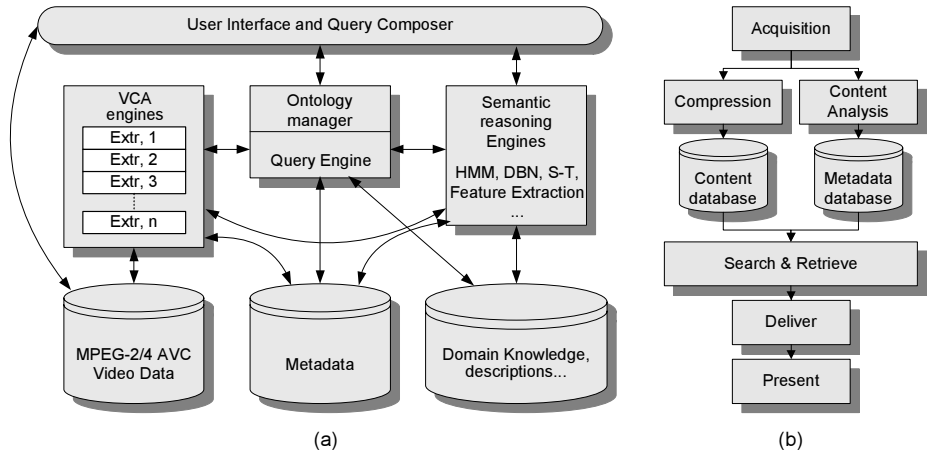


Fig. 1. General system architecture from Petrovic *et al.* (a) and from CANDELA (b)

- *Resource limitations in network and storage* - Communication between the various components of the system is via networks which have varying capabilities. Particularly, the last step in content transport to and from terminals (for acquisition, display and retrieval) is expected to be wireless. This can be either long-distance wireless networks like the cellular system or short-distance networks, like wireless LAN technologies or even Bluetooth. Bandwidth in these networks is expected to be limited and variable; scalable video coding and dedicated transmission protocols are required to guarantee real-time video streaming. Coding is also important for the storage requirements.
- *Content adaptation to terminal and user* - Content adaptation to terminal and user - Content will be retrieved through different types of terminals. In particular, terminal capabilities will vary greatly from powerful laptops to small-display telephones. In addition, the specific preferences of a user as summarized in her profile are taken into account. This requires content adaptation.
- *Interoperability and reliability in distributed processing* - As mentioned in the introduction, the components in Fig.1b may have a distributed realization. This holds in particular for the database. In addition, the functions of the system (like the VCA algorithms) may be realized through the cooperation of several devices. This requires these devices to be interoperable. In both cases, system reliability is a special concern.

2.1 Resource Limitations in Network and Storage

To reduce the storage requirements and communications requirements, video processing is adopted. To achieve a high fidelity content analysis, the

encoding and decoding stages should limit the loss of visual quality. Selection of the video coding format and the transmission method is based on requirements from the application, the terminal characteristics (e.g. camera/display resolution and computational capacity), the network connection, and the available storage capacity. Both non-scalable and scalable coding technologies have been studied. As a result, H.264/AVC and MPEG-4 simple profile (SP) are identified as the most important non-scalable coding standards, whereas AVC scalable video coding (SVC) looks very promising for the scalable solution. The non-scalable coding technologies do not support real adaptive behaviour on the fly. The non-scalable encoder typically is initialized to produce a fixed frame rate and frame size. Only the bit rate can be adjusted during the encoding process. MPEG-4 SP is found to be the best solution for low computational capability terminals. H.264/AVC coding requires much more computations and therefore hardware

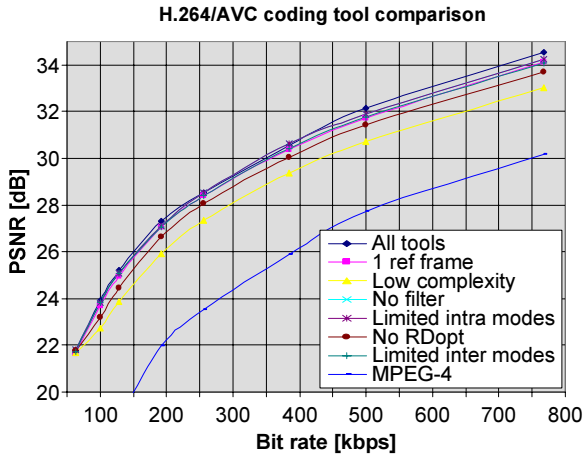


Fig. 2. Rate-distortion curves for different MPEG-4 AVC tools

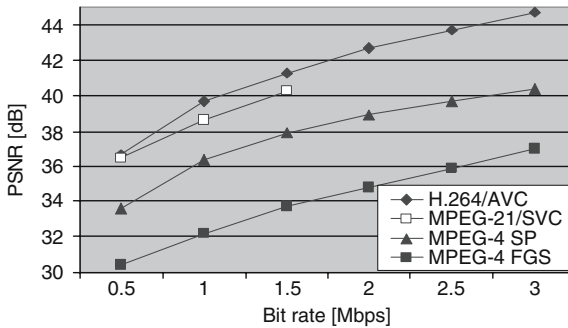


Fig. 3. Rate-distortion curves for scalable and non-scalable codecs

accelerators are often needed. Software solutions can be used for small frame sizes while the computational load of encoding can be reduced by limiting the coding features (tools) that may be exploited according to the standard (see Fig. 2). Scalable coding enables true adaptation to the network conditions by changing frame rate, size and quality. The MPEG-4 fine granularity scalability (FGS) includes tools for adaptation, but the nature of the scalable parts in the video stream leads to considerable bandwidth overhead. A more promising solution is offered by the the MPEG-21/SVC, which shows a superior rate-distortion behaviour compared to MPEG-4 FGS (see Fig. 3). However, the computational requirements might be too high for mobile terminals.

2.2 Content Adaptation to Terminal and User

The vast choice of user terminal types has created a new challenge for information service providers. Typically, the multimedia-oriented web services provide transportation, taking the different capabilities of the user terminals and network connections into account. The user terminals can vary from mobile phones to personal communicators, set-top boxes and personal computers. Three issues are important in order to provide appropriate content and layout to users. Firstly, the system needs to be aware of the user terminal capabilities and configuration. Secondly, it has to know the current state of the network connections. These two are used to select the right scalable coding and transport protocols. Thirdly, it should know the personal details of the user. It is important this type of information is handled automatically, without the need for manual interference. The web page content is automatically adjusted for transmission to a user client, depending on the current state. This minimizes the need for content creation on the author side as the same content can be used on a variety of user terminals. In CANDELA systems, the user interfaces for searching and browsing the video storage are generated dynamically by adapting the content according to the user terminal profile. This feature is provided by using XSLT stylesheets to adapt the MPEG-7/XML content descriptors. This approach gives a possibility to have suitable web content and layouts for numerous type of user terminals.

2.3 Interoperability and Reliability of VCA Applications

Database Architecture

Both metadata and video data have to be stored in the database, requiring the ability to download and stream data from and to the database (DB). From a logical point of view, this DB should be a single entity, however physically it should be distributed. This offers scalability of the system in data volumes and in terms of number of users.

The underlying DB technology is based on the Solid BoostEngine: a relational database server with SmartFlow data distribution capabilities [2]. Essentially, the DB is a small-footprint relational database that provides typical

functionality, such as SQL transactions, multi-user capabilities, and automatic data recovery. The server is embeddable in multiple HW/OS environments allowing cross-platform data distribution and supports API standards like ODBC and JDBC.

On top of the server there is a schema-dependent metadata abstraction layer. This layer contains an extendable set of object-oriented interfaces to support metadata in different formats, i.e. UPnP DIDLite and MPEG-7. The underlying object-oriented data model performs the mapping onto the relational model, thereby allowing access to the data both via high-level object-oriented and directly via more flexible relational (SQL) APIs.

The Solid DB management system (DBMS) has two separate storage methods: one for in-memory and another more traditional for on-disk based storage. Both methods can be used simultaneously to manage both *regular data*, which fits to limited-length database table columns, and *blob data*, which comprise large binary chunks of data. Because of this separation, the binary data can be handled more efficiently than alternatively storing them in regular files. Moreover, it is beneficial to keep all data within the database: only one API is needed to access all data; it enables to combine access to all content and metadata in the same queries; all data can be treated transactionally; all data of the system is behind the access control mechanism of the DBMS; and all data can be distributed across the databases of the system and various devices using a unified data-synchronization mechanism.

Distributed VCA Processing

For at least one of the applications domains, viz., the home multimedia, there is a natural trend towards distributed processing. There is a rapid growth of devices with storage and VCA capabilities aboard such as voice recognition or elementary video analysis. The concept of ambient intelligence, in which devices share their assets to realize a user-friendly and adaptive environment leads to the requirement of cooperation of all these devices. VCA and other applications are then realized through the cooperation of these local processing and storage capabilities. Therefore, we have developed a distributed platform for real-time VCA processing in the context of home networks. The focus in this design was on the separation between basic signal and video processing functions on the one hand and their composition into an application (e.g., commercial or genre detection, face recognition) on the other hand. The basic functions are available as services on the network in the way a standard like UPnP (Universal Plug 'n Play) defines them.

The platform naturally supports the mobility of services through dynamic and late binding. However, this also means that applications must be able to deal with services that disappear. More generally, the distribution leads to an increased number of points of failure. Particular attention has been paid to reliability through monitoring and signalling faulty behavior and subsequently taking corrective action, for example, by restarting a service on a different

device. This is in fact similar to the reliability concerns in the distributed data base.

3 Video Content Analysis

To provide the metadata for content-based video retrieval, annotation of the video content is required. In CANDELA, we aim at automatically analysing the video content to facilitate this annotation. The metadata is then stored in a multimedia database associated with the video data. The following describes application-specific VCA, addressed within the CANDELA project.

3.1 Medical Systems

The majority of colorectal cancers start as benign polyps that take many years to develop. The detection and removal of these polyps drastically reduces the number of cancer related fatalities. A CT scanner can be used to create three-dimensional images of the colon and detect polyps in an early stage. A typical CT volume consists of 500-1000 slices of each 512x512 pixels, resulting in 250-500 MByte of data. Radiology departments are challenged to cope with this data explosion while at the same time improving the quality of diagnosis. Methods were developed to assist visualization, extraction and navigation of the colon structure, but are still not able to perform a fully automated analysis. Related research in this field can be found in [3] to [6].

Three steps can be identified in the automated polyp detection schemes. First, to reduce computation time and limit the search space the colon wall is extracted. Then, for each voxel (3D pixel) in the colon wall, features are calculated that describe the local shape. Curvature of the colon surface is a powerful feature to distinguish the polyps from flat or ridge-like structures. In a next step, polyp-like features are clustered to obtain polyp candidates [7]. In this step the scheme yields a high false positive rate (ten to a few hundred per dataset), whereas the sensitivity is 90% or better. To reduce the false-positives, further content analysis is performed. The resulting features are subsequently used by a statistical classifier that is manually trained with a set of expert annotated polyps [8][9].

3.2 Surveillance

Video content analysis in the developed surveillance application comprises the detection of moving objects to provide automatic recognition of abnormal events (e.g. abandoned luggage at an airport) [10]. Objects can be detected by taking the difference between input frames and the scenes background [11]. By tracking objects over multiple video frames, trajectories and a simple description of events can be created.

One of the critical research areas that is gaining more attention is the validation of such VCA systems [12]. Quite some work on validation has been done,

however, standardization is required for viable validation and benchmarking. To stimulate the research in this area, validation sequences from CANDELA have been made publicly available.

Basically, the video content analysis (VCA) modules in the system analyze incoming video frames and segments the images into a static background and moving foreground objects. In addition, the objects are tracked over time, giving each object a unique object Id. Summarizing, for each frame and for each object the VCA modules output the location, the bounding box and an Id. The following subsection explain how higher-level analysis is applied to increase the semantic level of the video descriptions.

Metadata Analysis

Figure 4a shows an example of how the above-mentioned VCA outputs the results over the lifetime of an object. This format is not suitable for storage and retrieval. Therefore, the system contains separate Metadata Analysis modules (MDAs) that convert the frame-based descriptions from the VCA into object-based descriptions to remove redundancy and to provide a data format that is more suitable for retrieval (see Subsection 4.1). Notice that human reasoning is more object-oriented and hence search queries of this type are preferred. The conversion reduces the set of trajectory points by Piecewise Linear Approximation (PLA) [13] and the Piecewise Aggregate Approximation (PAA) [14]. This results in a more compact description of the trajectory. The

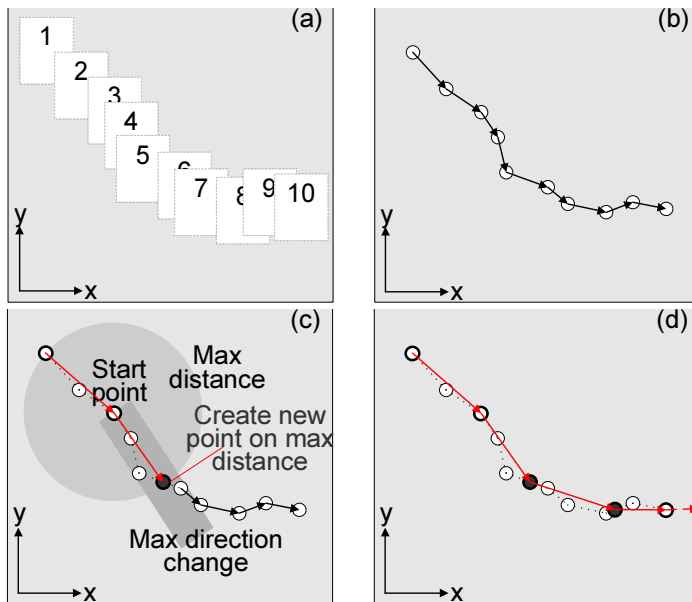


Fig. 4. Bounding box for each video frame (a), location for each video frame (b), filtering locations over time (c), original and filtered locations (d)

conversion is applied for each new object that is detected in the scene. The location of the object is defined by the center of the bounding box (see the example in Figure 4a and 4b). When the conversion algorithm decides that the location of the object at the current frame is relevant, it is stored into the database. The relevance is determined by two criteria. Firstly, the maximum distance between two location points from the conversion and secondly, the maximum deviation of the direction in which the object is moving. Both criteria are visualized in Figure 4c. When the conversion engine decides that any of these two criteria are exceeded, a new trajectory point is generated and stored. To compute the location of this new point, interpolating between the current and previous frame is applied. This algorithm is continued until the object has disappeared. The results from the conversion are shown in Figure 4d.

Perspective Transformation

As mentioned before, each object is described by its location and the bounding box, denoted in the number of pixels. However, units of pixels are not desired due to the perspective distortion that is introduced by the 2-D image acquisition of the 3-D world. Note for example that the object size decreases when the object moves further away from the camera. Therefore, in order to use the location and bounding box information for intuitive search queries, the pixels coordinates are transformed to real-world size coordinates. This requires manually or automatic calibration of the camera [15] [16], i.e. the height of the camera and the distance to two points in the scene have to be determined (see Figure 5). Subsequently, perspective transformation can be applied to compute the real-world sizes of detected objects.

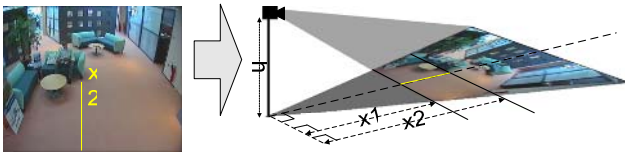


Fig. 5. Pixels to meters, using perspective transform

Classification

After deriving the real-world sizes, the metadata processing applies straightforward classification by using *a priori* knowledge on the typical sizes of persons, cars, trucks, and smaller objects like rabbits or suitcases. Moreover, the trajectory coordinates are used to determine the speed of the objects. This enables a more accurate classification, since it is unlikely that for example a person walks at a speed of 40 km/h. Hence, the combination of these properties enables some mean of higher level reasoning.

3.3 Home Multimedia

Terabytes of storage capacities, millions of MIPS (Million Instructions Per Second) of processing power and gigabits of bandwidth become a reality in consumer homes [17]. The resulting explosive growth of digital multimedia assets in the inter- and broadband connected home creates the need for advanced methods of browsing, navigating, retrieval and content management. Content descriptors [18] enable these advanced methods and extend their capabilities. In CANDELA several VCA solutions have been elaborated and benchmarked, such as a multimodal scene segmentation algorithm [19] [20]. Moreover, face detection, face localization and identification have been researched to enable semantic meaningful high-level search queries on consumer devices [17].

In addition to the VCA algorithms, also advanced distributed-VCA architectures, protocols and technologies based on a network of legacy CE devices have been investigated in CANDELA, utilizing distributed processing power as available in In-Home networks. [21].

3.4 Application Adaptivity

Jan, Egbert

This part should describe some means how the system can automatically apply the correct analysis. Does this hold for medical?

4 Retrieval Application

Content-based video retrieval is in essence the paradigm of web search engines extended to multimedia data. Basically, video data is annotated with metadata to describe its semantic contents, thereby allowing to search and retrieve videos containing the types of actions, objects, behavior, scenery etc. To enable the system to search for these properties, metadata from the VCA needs to be processed to match the users search queries. Moreover, to enable the system to truly "understand" the content, ontologies are provided. Basically, the ontologies mainly define the relation between descriptors and search request. These ontologies offer the ability to make the queries adaptive to the search request by automatically learning more relations.

4.1 Surveillance

From a user point of view, it is desirable to search through the video database without any additional expert knowledge. For the surveillance application, we have amongst others looked at search queries that are related to the trajectories (motion paths) of the objects [22][23]. Typically, such trajectory data is conveyed by the VCA algorithm on a frame-basis. However, as explained in Subsection 3.2, this format is not suitable for efficient storage in a database (DB)

nor for matching the trajectory with a query request. Let us explain by example. At the Graphical User Interface (GUI) side, a user is able to sketch a trajectory in the plane of the camera-view to search for all objects with a similar trajectory pattern. However, a fundamental problem is that all trajectory points of all objects in the database have to be examined to find the correct matches. Therefore, a fast but still accurate method is required to reduce this computational burden.

Three different challenges can be distinguished. Firstly, the definition of the *data representation* to model trajectory data for efficient indexing in the database. This was discussed in Subsection 3.2. Secondly, which *database indexing structure* is used that provides fast searching, without scanning the whole database? Thirdly, we need to define the *matching model*: Which metric is going to be used as a distance (quality) measure between trajectories? As a requirement, the chosen data structure should support different types of queries: retrieval of parts (sub-trajectories) of the trajectory data that match with the sketched query; retrieval of objects that crosses a user drawable line and; retrieval of objects in a selected interesting area. The following will describe the indexing and matching challenges separately.

Database Indexing Structure

After studying several database indexing structures that can store spatial data, R-tree variants [24] seem to fit our requirements best. Many geographical information systems (GIS), already extensively use R-trees for similar applications [25]. Spatial indexing is done by grouping the approximate trajectory representations into sub-trails and representing each of them with a minimum bounding rectangle (MBR). For our application, a special variant of R-trees (R*-tree) is

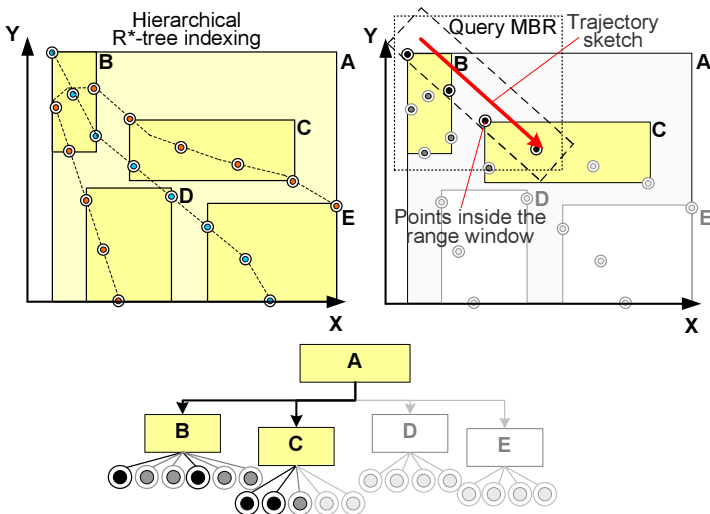


Fig. 6. Hierarchical storage of trajectory data

used to store the MBRs in an hierarchical way (see the left part of Figure 6). After the user sketches a trajectory to search for, a window is placed over the drawn line segments to define a distance range. This range defines the area of the trajectory points to search for. After this process, the hierarchical R*-tree filled with trajectory data is traversed for each query MBR (see Figure 6 for an example where two trajectories are present in the R-tree).

Matching Model

For matching the sketched line(s) with the trajectories in the database, two different metrics are adopted: the Euclidean point-to-line distance and the directional difference between the queried line and the stored line segments. If the sketch trajectory query contains more than one MBR, the matching is first applied to each MBR. To enable a Google-like ranking of the retrieved objects and to provide preliminary results for fast feedback to the user, the ranked results from each MBR query are combined into one global result set. Therefore, for each two MBR query result sets, a rank-join algorithm is executed that joins the trajectory points from the two sets. Finally, one large result set, ranked in the order of similarity, is left that contains all trajectories that match with the user sketch. The ranking phase in the rank-join algorithm is adaptive to the size of the MBR and its number of processed points [26].

4.2 Ontologies

A common definition of ontology in information technology is the mean to provide a shared and common understanding of a domain that can be communicated between people and heterogeneous, widely-spread application systems [27]. Obviously, such a shared understanding would be extremely helpful for an effective annotation and retrieval of video content in CANDELA as well. This requires the capture of concepts to describe all aspects of all CANDELA users, which are all different. Obviously, such universal ontology is not feasible, but general categories of concepts can be identified, such as: home, work, hobbies, family etc. In addition, one may consider an additional set of descriptions: who is acting; where is the action happening; what type of action; etc.

Within CANDELA a set of top-level ontological classes are predefined that take above-stated descriptors into account (e.g., persons, places, objects, events), as well as several properties (e.g., name, location, sex). The latter are useful for interrelating and describing instances of these classes (e.g., a specific person or place). Subsequently, the classes and properties give a description of the video. The ontology manager is responsible for expanding the terms a user provides in a keyword search. Right before performing the keyword search on the metadata, the query engine consults the ontology manager to add the names of all classes, subclasses, and instances from the user's personal ontology. For example, if Anna supplies the term "child", the ontology manager consults her personal

ontology and adds the terms "Sofia" and "daughter" to the query, because these are names of all subclasses and the instances of these subclasses of the class "child". The benefit of this is that the query will now also consider videos that Anna has annotated with the terms "Sofia" or "daughter" but not with "child" explicitly. At a higher stage, the ontology can include rules that will adapt queries to the context of the user. By using information about the place, the time or the personal data of the user, obtained through mobile terminals such as mobile phones, the ontology could make the query more efficient. For example, when Anna is at work and queries the word "trip", unless otherwise stated, the result of the query will give priority to videos from working trips. More information can be found in [28].

5 Conclusion

Secondly, the paper will elaborate on the application-specific content analysis in Section 3. The last subsection, we will show how the system can automatically adapt the content analysis to the needs of the application domain. Thirdly, Section 4 shows how the retrieval application differs per application domain. One of the subsections explains how ontologies can be exploited to introduce human-like reasoning into the system, i.e. provide the relation between human-natural search queries and the content descriptors.

Although many technologies exist for content-analysis, storage, retrieval and delivery, the integration of these technologies enable fast and efficient human interaction with distributed systems. In particular, the high semantic level of interaction and the physically distributed means for storage and delivery are considered as novelties. This paper particularly highlights the adaptive aspects.

Concerning the system architecture we distinguished the following. Firstly, we studied scalable coding to adapt video transmission to network conditions. The study showed that conventional scalable coding such as MPEG-4 FGS results in considerable bandwidth overhead, whereas MPEG-21 shows far more better results, but pays its price in computational requirements. Secondly, the adopted database technology also fits with the theme of adaptive retrieval, since it offers automatic recovery of malfunctioning components. Although the database technology for distributed storage and delivery is generic, dedicated features were added such as: an MPEG-7 interface, the storage of arbitrary-size binary data (blobs), and streaming of multimedia content. Thirdly, we identified adaptive content delivery. User profiles and identification of terminal capabilities enable the system to adapt the content accordingly. Finally, we identified the service-based architecture as an adaptive aspect of the system architecture. Capabilities of the system are automatically adjusted to the availability of the audio-video components. To service a specific functionality, the corresponding components should be available. This offers quality of service, but also provides robust operation in case of malfunctions.

Although we consider the system architecture to be the main contribution to adaptive retrieval, the content analysis algorithm itself are adaptive by nature, since the processing highly depends on the content of the signal. By exploring several application domains we can conclude that general analysis technology seems impossible. However, by gather several VCA modules in a service-based architecture, the system could automatically adapt to the application domain by requesting application-specific analysis services.

The last adaptive aspect concerns the retrieval itself. Besides the functionality to analyze content and store everything, a query engine is needed to access the desired content. This part of the system can considerably gain intelligence by adaptively updating the ontologies to the search queries.

Although the potential of CANDELA-like has been indicated, the development is still in its infancy. Hence, many research areas have been identified for future exploration.

Acknowledgements

Several authors are denoted on the paper. However, the work was performed by many more people. We would like to thank all members of the CANDELA project.

References

1. M. Petkovic and W. Jonker, *Content-Based Video Retrieval, A Database Perspective*, Multimedia Systems and Applications, Vol. 25. Springer, 2003.
2. www.solidtech.com/library/AutonomicDataManagement.pdf.
3. R.E. van Gelder *et al.*, "Computed tomographic colonography compared with colonoscopy in patients at increased risk for colorectal cancer," *Gastroenterology*, vol. 27, no. 1, pp. 41–48, July 2004.
4. M. Medved *et al.*, "Segmentation and segment connection of obstructed colon," in *Proc. of the SPIE - Medical Imaging 2004*, Febr. 2004, vol. 5370.
5. A.H.de Vries *et al.*, "Feasibility of automatic prone-supine matching in ct-colonography: precision and practical use," in *Proc. of 5th Int. Symp. on Virtual Colonoscopy*, Oct. 2004.
6. J. Florie, "Automatic cleansing for ct colonography using a three material transition model," in *Proc. of 5th Int. Symp. on Virtual Colonoscopy*, Oct. 2004.
7. J. Nappi and H. Yoshida, "Feature-guided analysis for reduction of false positives in cad of polyps for computed tomographic colonography," *Med Phys*, vol. 30, no. 7, pp. 1592–1601, 2003.
8. A. K. Jerebko *et al.*, "Computer-assisted detection of colonic polyps with ct colonography using neural networks and binary classification trees," *Med Phys*, vol. 30, no. 1, pp. 52–60, 2003.
9. S. B. Gokturk *et al.*, "A statistical 3-d pattern processing method for computer-aided detection of polyps in ct colonography," *IEEE Trans Med Imaging*, vol. 20, no. 12, pp. 1251–1260, 2001.

10. P. Merkus *et al.*, "Candela - integrated storage, analysis and distribution of video content for intelligent information system," in *Proc. of the Euro. Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, EWIMT*, Nov. 2004, pp. 25–26.
11. X. Desurmont *et al.*, "Image analysis architectures and techniques for intelligent surveillance systems," *IEE Proc. - Vision, Image and Signal Processing*, 2005.
12. X. Desurmont *et al.*, "Performance evaluation of real-time video content analysis systems in the candela project," in *Proc. of the SPIE - Real-Time Imaging IX*, Jan. 2005.
13. C.B. Shim and J.W. Chang, "Spatiotemporal compression techniques for moving point objects," in *Advances in Database Technology - EDBT 2004: Proceedings of the 9th International Conference on Extending Database Technology*. 2004, pp. 765–782, Springer-Verlag.
14. E.J. Keogh and M.J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*. 2000, pp. 122–133, Springer-Verlag.
15. S. Deng, Y. Yang, and X. Wang, "A calibration method using only one plane for 3d machine vision," in *16th International Symposium on Electronic Imaging, Storage and Retrieval Methods and Applications for Multimedia, SPIE 2004*, Jan. 2004.
16. J.R. Renno, J. Orwell, and G.A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," in *The 13th British Machine Vision Conference*, Sept. 2002, pp. 607 – 616.
17. J. Nesvadba *et al.*, "Face related features in consumer electronic (ce) device environments," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Oct. 2004, pp. 641–648.
18. J. Nesvadba *et al.*, "Comparison of shot boundary detectors," in *Int. Conf. for Multimedia and Expo, ICME*, June 2005, submitted for publication.
19. J. Nesvadba *et al.*, "Low-level cross-media statistical approach for semantic partitioning of audio-visual content in a home multimedia environment," in *Proc. IEEE Int. Workshop on Systems, Signals and Image Processing*, September 2004, pp. 235–238.
20. European ITEA program, *Candela Deliverable D1.1B, State-of-the-art Report, Appendix: AV corpus*, March 2004, www.extra.research.philips.com/euprojects/candela/deliverables/candela-wp1-d1-signoff.pdf.
21. F.de Lange and J. Nesvadba, "A hardware/software framework for the rapid prototyping of multimedia analysis systems," in *Proc. of the Workshop of Image Analysis for Multimedia Interactive Systems*, April 2005, submitted for publication.
22. Y. Yanagisawa, J. Akahani, T. Satoh, "Shape-based similarity query for trajectory of mobile objects," in *Proc. Lecture Notes in Computer-Science on the 4th int. conf. on Mobile Data Management*, 2003, vol. 2574, pp. 63–77.
23. S. Satoh K. Aizawa, Y. Nakamura, "Sketchit: Basketball video retrieval using ball motion similarity," in *Advances in Multimedia Information Processing - PCM 2004: Proceedings of the 5th Pacific Rim Conference on Multimedia*, Tokyo, Japan, October 2004, vol. 3332, p. 256, Springer-Verlag GmbH.
24. Y. Manolopoulos *et al.*, "R-trees have grown everywhere," 2003.

25. R.K.V. Kothuri, S. Ravada, and D. Abugov, "Quadtree and r-tree indexes in oracle spatial: a comparison using gis data," in *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2002, pp. 546–557, ACM Press.
26. I.F. Ilyas, W.G. Aref, and A.K. Elmagarmid, "Joining ranked inputs in practice," in *VLDB*, 2002, pp. 950–961.
27. D. Fensel and M.L. Brodie, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, 2001.
28. A.Sachinopoulou *et al.*, "Personal video retrieval and browsing for mobile users," in *Proc. of the SPIE - Multimedia on Mobile Devices*, Jan. 2005.

Interactive Retrieval of Video Sequences from Local Feature Dynamics

Nicolas Moënne-Loccoz, Eric Bruno, and Stéphane Marchand-Maillet*

Viper Group, Computer Vision and Multimedia Lab, University of Geneva - 24,
rue du Général Dufour, 1211 Geneva 4, Switzerland
Nicolas.Moenne-Loccoz@cui.unige.ch

Abstract. This paper addresses the problem of retrieving video sequences that contain a spatio-temporal pattern queried by a user. To achieve this, the visual content of each video sequence is first decomposed through the analysis of its local feature dynamics. Camera motion of the sequence, background and objects present in the captured scene and events occurring within it are represented respectively by the parameters of the estimated global motion model, the appearance of the extracted local features and their trajectories. At query-time, a probabilistic model of the visual pattern is estimated from the user interaction, captured through a relevance-feedback loop. We show that the method permits to efficiently retrieve video sequences that share, even partially, a spatio-temporal pattern.

1 Introduction

Content based retrieval of video sequences within large collections is mainly a matter of similarity estimation. For genericity purposes, the similarity should rely on a representation of video sequences that makes as few as possible assumptions about their content and that captures spatial and temporal sub-part of it. The associated index structure should be scalable, and therefore the complexity of the representation limited. Finally, the similarity should be adaptively defined by the user through an interactive online procedure so that it may finally characterize the queried spatio-temporal pattern.

Local features are atomic parts of an image that cover its most salient content. They have properties that make them well-adapted to define the dynamics of a video sequence. Also, their extraction is efficient and makes very few assumption about the content. Hence, they are particularly suitable to index video sequences for content-based retrieval of spatio-temporal patterns. Such features have been extensively used for image retrieval [14, 21] and tend to be more and more used for object recognition [3, 10] and video indexing [18, 20]. Different approaches have been proposed to retrieve visual patterns from local features. First, in *model-based approaches*, a model of local feature appearances, relative scales and positions is estimated from a set of positive and negative examples. The

* This work is funded by EU-FP6 IST-NoE SIMILAR (www.similar.cc) and the Swiss NCCR IM2 (Interactive Multimodal Information Management).

most informational features are automatically selected to be the support of the model. That way, occurrences of learnt objects may be detected from the local features. But the estimation of such parametric models (e.g. constellation model [11]) requires lot of training examples and have an unmanageable algorithmic complexity. The *matching* approaches compute a score for each indexed image (or video frame) according to the number of features it shares with the query. This similarity may be enhanced by estimating the geometric transformation between groups of features. Hence, images containing the same rigid objects and/or captured within the same background as the query may be accurately retrieved. However, the dynamics of features are not taken into account and also the complexity of the process prevents its use for large collections. Finally, Sivic *et al* [20] have proposed a more information retrieval oriented approach. They construct a vocabulary of local feature appearances and use text retrieval similarity measure to estimate the relevance of indexed video sequences. While this permits to index a large amount of sequences and performs well in terms of retrieval, it first does not consider the dynamics inherent to video sequences. Also, relevant local features are not automatically selected ; instead the user has to provide as a query a set of local features, chosen within some images. Furthermore, authors do not discuss the fact that text retrieval methods may not be adapted to this problem as local features in a video sequences do not carry the same information as a term in a text document.

The approach proposed in this paper uses the local feature appearances and trajectories to index the content of video sequences. Based on this index, a probabilistic model of spatio-temporal patterns is proposed. It may be efficiently estimated from only some positive and negative examples provided interactively by the end user. That way, relevant local feature appearances and trajectories are automatically and efficiently selected to provide the user with an accurate estimation of the queried spatio-temporal pattern.

The second section of the paper presents the local feature extraction process and the estimation of their trajectories that permit to estimate the camera motion model. Section 3 shows how video sequences are indexed from this information, for efficient processing of queries. Section 4 details the model of video sequences visual content and how several such models are interactively estimated to retrieve relevant sequences. Finally, the section 5 presents some evaluations of the proposed method that demonstrate the ability of our model to capture the queried visual content and the accuracy of its interactive estimation.

2 Visual Content Representation from Local Feature Dynamics

Local features W define a local decomposition of the image space V that differs from a segmentation of V because it does not covers it entirely. Furthermore, local features do not characterize visual entities but instead regions with a salient signal structure. Hence, they have been originally proposed to estimate the geometric transformations between images. More recently they have been used

for visual information retrieval and object recognition because of their robustness, their repetitivity and their information content.

2.1 Local Features Extraction

A number of different local features have been proposed in the literature. The most noticeable are the Förstner-Harris corners [5, 6], the Lindeberg Laplacian blobs [12], the Wavelet-Based salient points [21] and the regions of maximum entropy [8]. Among them, the interest points proposed by Schmid *et al* [14] which are Harris-Laplacian corners, and their affine invariant formulation [15] may be considered as the most reliable ones. In our work, we have preferred to use the difference-of-Gaussian (doG) feature points proposed by David Lowe [13] which are an estimation of the Laplacian blobs of Lindeberg, enhanced with some heuristics. These feature points are less robust in their position than the interest points, but as their extraction is very efficient, and as they correspond to the main blob-like structures of the image, they provide a suitable enough representation of the visual content. DoG feature points correspond to local maxima of the estimated Laplacian $\hat{L}(w, s)$ (estimated by a difference of Gaussians) in both the image space V and the scale space S ($s \in S$) of the frame. For a given video frame, the set W of the detected feature points $w \in V$ are those satisfying the following constraints :

$$\left(\hat{L}(w, s) > \hat{L}(w, s - 1)\right) \wedge \left(\hat{L}(w, s) > \hat{L}(w, s + 1)\right) \quad (1)$$

$$\left(\hat{L}(w, s) > \tau_{\hat{L}}\right) \wedge \left(\hat{L}(w, s) > \hat{L}(N_{w,s}, s)\right) \quad (2)$$

where $\tau_{\hat{L}}$ is a threshold and $N_{w,s}$ stands for the local neighborhood of the point w at its characteristic scale. The local maximization of the Laplacian in the scale space (equation (1)) is justified by the scale-space theory [12]. The idea is that local maxima represent points in the scale space where structures disappear. The other constraints (equation (2)) aim at considering only the most stable structures. The feature points W are represented using the *SIFT* descriptor that has been identified as the most reliable one [16]. This descriptor is a spatial histogram of the orientation and strength of the gradient within $N_{w,s}$. As the gradient orientations are normalized w.r.t. the main orientation, the descriptor is scale and rotation invariant.

2.2 Local Feature Dynamics

Local features are robust to geometric and luminance transformations and they have a good repetitivity. These properties make them easy to match [19]. Hence, their trajectories may be obtained without performing a complex tracking process. The local feature dynamics permit to estimate the camera motion performed during the capture of the sequence, and the residual motion coming essentially from the moving objects. These informations are therefore important cues to defined similarity between video sequences.

Local Feature Trajectories. In order to obtain the local feature trajectories $W^{[t,t+1]} \in W^t \times W^{t+1}$, they are matched between the two consecutive frames t and $t + 1$. In other words, an injection $\pi_{t,t+1}$ between the two sets of points has to be found that minimizes the *matching distance* :

$$\pi_{t,t+1} = \arg \min_{\pi} \left(\sum_{i=1}^{|W^t|} d_{\text{sift}} \left(w_i^t, w_{\pi(i)}^{t+1} \right) \right) \quad (3)$$

where d_{sift} is the function returning the distance between the *SIFT* descriptors of two local features. The *Hungarian* algorithm [9] has been proposed to obtain $\pi_{t,t+1}$ in $O(|W^t|^3)$. However, for performance reasons, we use the greedy approach that estimate $\pi_{t,t+1}$ by iteratively choosing the match with the minimal distance. Using this matching algorithm along a video sequence \mathcal{S} , we obtain the set of local features trajectories $W_{\mathcal{S}}^{T_{\mathcal{S}}}$ where $T_{\mathcal{S}}$ is the temporal space of \mathcal{S} . Trajectories that have a length below a given threshold are removed because they correspond to either spurious features or to features that are not representative enough of the visual content of the sequence. In the sequel, we will use the notation $W_{\mathcal{S}}$ to represent the dynamic local features (local feature trajectories) extracted, tracked and filtered for the whole sequence \mathcal{S} .

Camera Motion Estimation. Given the set of trajectories $W_{\mathcal{S}}$, we estimate the most representative affine motion model that corresponds to the camera motion occurring during the sequence :

$$\mathbf{v}_{\mathbf{w}} = \begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix} \mathbf{w} + \begin{pmatrix} a_3 \\ a_6 \end{pmatrix} \quad (4)$$

To estimate the model parameters, we apply a *RanSaC* algorithm [4]. As the set of trajectories contains noise, the model is then smoothed by applying a *Tukey M-estimator* in a way close to the one presented in [22].

Local Feature Dynamics. Trajectories of local features have two components: the camera motion that is embedded in all trajectories and the feature absolute motions. In order to get the real dynamics of the local features, trajectories are compensated w.r.t. the camera motion. The features having null trajectory usually belong to the background of the scene or characterize static objects present in the scene. The other trajectories characterize moving objects, i.e. these trajectories are cues about the events occurring within the sequence.

3 Visual Content Indexing

The visual content of video sequences is represented by their camera motion and by the appearance and the compensated trajectory of their local features. Because of their complexity, video sequences cannot be directly indexed using this information. Effectively, hundreds of local features are extracted for each sequence. Even if this amount of information is reduced by filtering low scale

and short trajectory features, it remains unmanageable. Hence, from this information, a simpler index is defined that captures what is relevant for the model we propose in section 4.

3.1 Camera Motion

The global dynamic is represented by the six affine parameters (see equation (4)). In order to get a more informational representation of the camera motion the six affine parameters are mapped to the following vector :

$$\mathbf{C}_S = \begin{pmatrix} \text{Pan} \\ \text{Tilt} \\ \text{Zoom} \\ \text{Rotation} \end{pmatrix} = \begin{pmatrix} |a_1| \\ |a_4| \\ \frac{|a_2+a_6|}{2} \\ \frac{|a_5-a_3|}{2} \end{pmatrix} \quad (5)$$

Hence, each video sequence \mathcal{S} is indexed according to its camera motion, by the vector \mathbf{C}_S that describes the occurrence of the different camera motions.

3.2 Local Feature Appearances

Local feature appearances is described by the mean of their *SIFT* descriptors along their trajectory. In order to efficiently index the video sequences according to their appearances we quantized their description, thus obtaining a fixed size vocabulary. However *SIFT* are 128-dimensional vectors, and they cannot be accurately uniformly quantized. Instead, *SIFT* descriptors are clustered using an algorithm that is able to handle a very large data set, because the number of extracted features cannot be handled directly (e.g. $> 1\overline{M}$ for the whole TRECVID 2003 collection). We use the *BIRCH* algorithm [23] that aims at constructing a tree (*Clustering Tree*) to reduce the size of the data set while maintaining its diversity. From this tree a classical clustering algorithm may be applied.

Applying the *BIRCH* algorithm, and removing the clusters having few components (≤ 2), a vocabulary \mathcal{A}_{Ω_S} of approximately 3500 feature models is obtained. Hence, the vector \mathbf{A}_S may be defined for every video sequence \mathcal{S} . It stores the number of occurrences of each feature model in \mathcal{S} :

$$\forall a_i \in \mathcal{A}_{\Omega_S}, \mathbf{A}_S(i) = |W_S^{a_i}| \quad (6)$$

where W_S^a is the subset of W_S such that for all $w \in W_S^a$, the *SIFT* signature of w belongs to the feature model $a \in \mathcal{A}_{\Omega_S}$.

3.3 Local Feature Dynamics

Local feature dynamics are described by the energy of their compensated trajectory, in the frame space V . Again, the local feature trajectories are clustered in order to obtain a vocabulary defining the most representative trajectories within the collection Ω_S . The set \mathcal{D}_{Ω_S} of 500 feature trajectory models is used to describe the local feature dynamics of each sequence \mathcal{S} . The vector \mathbf{D}_S captures the number of occurrences of each element of \mathcal{D}_{Ω_S} in \mathcal{S} :

$$\forall d_i \in \mathcal{D}_{\Omega_S}, \mathbf{D}_S(i) = |W_S^{d_i}| \quad (7)$$

where W_S^d is the subset of W_S such that for all $w \in W_S^d$, the trajectory of w belongs to the feature trajectory model $d \in \mathcal{D}_{\Omega_S}$.

4 Interactive Visual Content Retrieval

The visual content of a video sequence may be divided into the following four main entities :

- **Editing Effects:** camera motions with which the sequence has been captured.
- **Scene:** the background that has been captured along the sequence.
- **Objects:** the objects present in the scene during the capture of the sequence.
- **Events:** the actions of the objects occurring while the sequence was captured.

The information provided by the local features does not permit to distinguish static objects from the background. Hence in the proposed model, we consider only three different entities : camera motions (Editing Effects), local feature appearances (Scene and Objects) and local feature dynamics (Events). However, as the interactive estimation of the model is able to select among the local features those that are relevant, this simplification does not limit the expressiveness of the model.

4.1 Visual Content Model

A spatio-temporal pattern occurring in a video sequence is assumed to be defined by some camera motions, to occur within a given background and to contain some objects and some events. The background and the objects are characterized by a subset of \mathcal{A}_{Ω_S} (feature appearance models) and the events are characterized by a subset of \mathcal{D}_{Ω_S} (feature trajectory models). Figure 1 presents how such a pattern is modeled by a Bayesian network. The probability of a video sequence \mathcal{S}

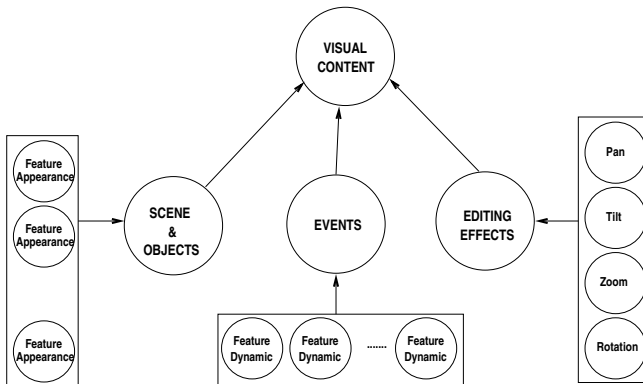


Fig. 1. Model of the visual content of a video sequence

to contain a specific visual pattern is computed using the Bayesian formulation from its index vectors \mathbf{C}_S , \mathbf{A}_S and \mathbf{D}_S :

$$\begin{aligned}
P(\text{Visual Content}|\mathcal{S}) &= P(\text{Visual Content}|\mathbf{C}_S, \mathbf{A}_S, \mathbf{D}_S) \\
&= P(\text{Editing Effects}|\mathbf{C}_S)P(\text{Scene Objects}|\mathbf{A}_S)P(\text{Events}|\mathbf{D}_S) \quad (8) \\
&= \frac{P(\mathbf{C}_S|\text{Editing Effects})P(\mathbf{A}_S|\text{Scene Objects})P(\mathbf{D}_S|\text{Events})}{P(\mathbf{C}_S)P(\mathbf{A}_S)P(\mathbf{D}_S)}
\end{aligned}$$

Priors about the *Editing Effects*, the *Scene Objects* and the *Events* are not taken into account since they are constant for all video sequences we aim at comparing.

Editing Effects Model. The model is the probability of amount of a camera motion in the pattern :

$$\frac{P(\mathbf{C}_S|\text{Editing Effects})}{P(\mathbf{C}_S)} = \frac{G(\mathbf{C}_S|\theta_{S^q})}{\prod_{i=1}^4 P(\mathbf{C}_S(i))} \quad (9)$$

The likelihood is estimated by a Gaussian density function $G(\mathbf{C}_S|\theta_{S^q})$. The parameters $\theta = (\mu, \Lambda)$ are estimated from S^q under the assumption that the different motions are independent.

The evidences $P(\mathbf{C}_S(k)), \forall k \in [1..4]$ are estimated from the whole collection Ω_S .

Scene and Objects Model. The model is the probability of occurrence of a set of local features $\mathcal{A}^q \subset \mathcal{A}_{\Omega_S}$ in the pattern :

$$\frac{P(\mathbf{A}_S|\text{Scene Objects})}{P(\mathbf{A}_S)} = \frac{\prod_{a_i \in \mathcal{A}^q} P(\mathbf{A}_S(i)|\text{Scene Objects})}{\prod_{a_i \in \mathcal{A}^q} P(\mathbf{A}_S(i))} \quad (10)$$

For a set S^q , \mathcal{A}^q corresponds to the different elements of \mathcal{A}_{Ω_S} occurring at least once in S^q . The likelihoods $P(\mathbf{A}_S(k)|\text{Scene Objects}), \forall a_k \in \mathcal{A}^q$ are estimated from S^q by :

$$P_{S^q}(\mathbf{A}_S(k)|\text{Scene Objects}) = \frac{\sum_{S_i \in S^q} \sum_{j=1}^{\mathbf{A}_S(k)} H(\mathbf{A}_{S_i}(k) - j)}{|S^q|} \quad (11)$$

where

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

is the unit step function.

The evidences $P(\mathbf{A}_S(k)), \forall a_k \in \mathcal{A}^q$ are estimated in the same manner, from all sequences of the collection Ω_S .

Events Model. The model is the probability of occurrence of a set of local feature trajectories $\mathcal{D}^q \subset \mathcal{D}_{\Omega_S}$ in the pattern :

$$\frac{P(\mathbf{D}_S|\text{Events})}{P(\mathbf{D}_S)} = \frac{\prod_{d_i \in \mathcal{D}^q} P(\mathbf{D}_S(i)|\text{Events})}{\prod_{d_i \in \mathcal{D}^q} P(\mathbf{D}_S(i))} \quad (13)$$

For a set \mathcal{S}^q , \mathcal{D}^q corresponds to the different elements of $\mathcal{D}_{\Omega_{\mathcal{S}}}$ occurring at least once in \mathcal{S}^q . The likelihoods $P(\mathbf{D}_{\mathcal{S}}(k)|\text{Events})$, $\forall d_k \in \mathcal{D}^q$ are estimated from \mathcal{S}^q by:

$$P_{\mathcal{S}^q}(\mathbf{D}_{\mathcal{S}}(k)|\text{Events}) = \frac{\sum_{\mathcal{S}_i \in \mathcal{S}^q} \sum_{j=1}^{\mathbf{D}_{\mathcal{S}}(k)} H(\mathbf{D}_{\mathcal{S}_i}(k) - j)}{|\mathcal{S}^q|} \quad (14)$$

The evidences $P(\mathbf{D}_{\mathcal{S}}(k))$, $\forall d_k \in \mathcal{D}^q$ are estimated in the same manner, from all sequences of the collection $\Omega_{\mathcal{S}}$.

4.2 Interactive Query Model Estimation

A user query is defined to be a spatio-temporal pattern that occurs within the indexed collection $\Omega_{\mathcal{S}}$. In order to integrate the user knowledge about the query, we have adopted the relevance feedback loop [1] as the interaction protocol. In this protocol the user chooses a set of positive examples $\mathcal{S}^+ \subset \Omega_{\mathcal{S}}$ and a set of negative examples $\mathcal{S}^- \subset \Omega_{\mathcal{S}}$. \mathcal{S}^+ define the query (generative examples) that has to be retrieved whereas \mathcal{S}^- define some patterns that are close but distinct to the query (discriminative examples). It should be noted that generally the examples \mathcal{S}^- do not define exactly the negative query, i.e. $\Omega_{\mathcal{S}} - \{\mathcal{S} \ni \text{Query}\}$, instead the user tends to select \mathcal{S}^- as discriminative surrounding false examples. Hence, at each step, from $\mathcal{S}^q = \mathcal{S}^+ \cup \mathcal{S}^-$, the ranking of an indexed video sequence \mathcal{S} is not the Bayesian decision, but is defined by :

$$P_{\mathcal{S}^q}(\text{Query}|\mathcal{S}) = P_{\mathcal{S}^+}(\text{Query}|\mathcal{S}) (1 - P_{\mathcal{S}^-}(\overline{\text{Query}}|\mathcal{S})) \quad (15)$$

The user may iteratively refine the query by enriching \mathcal{S}^q from the returned ranked list of video sequences.

5 Experimental Evaluation

For the evaluation of the proposed algorithm, we use a subset of the TRECVID collection which contains hundreds of hours of *CNN* and *ABC* news broadcasts. The videos are first segmented using the algorithm presented in [7], in order to get sequences having a consistent visual content. We obtain a set $\Omega_{\mathcal{S}}$ of 2000 video sequences for which the representative local features are extracted. More than 60,000 feature descriptors are indexed in the multimedia document management framework presented in [17]. We have manually annotated $\Omega_{\mathcal{S}}$ in order to get a reliable ground-truth for the evaluation protocol.

To evaluate the retrieval capabilities of the algorithm, a batch of user queries are simulated. Initial sets \mathcal{S}^+ and \mathcal{S}^- are randomly chosen. Sequences of the collection $\Omega_{\mathcal{S}}$ are ranked according to the similarity measure defined by the corresponding estimated models. The sets \mathcal{S}^+ and \mathcal{S}^- are then enriched with examples randomly selected from the top-100 elements of the ranked list in order to mimic the behavior of a real user. The process is iterated until \mathcal{S}^+ cannot be enriched anymore or until a maximum number of iterations is reached. For each annotated concept the simulation is performed 50 times to collect statistics about the retrieval behavior of the algorithm.









<i>Concept</i>	ABC Anchor	CNN Anchor	ABC Studio	CNN Studio
<i>Frequency</i>	6.05%	4.60%	0.75%	5.35%
				
<i>Concept</i>	Basketball	Hockey	Car	USA Weather Map
<i>Frequency</i>	1.60%	0.75%	1.35%	1.65%
				

Fig. 2. Annotated concepts used in the experimentations: frequencies within the corpus and illustrative examples

Figure 2 presents the concepts considered in the experimentations we have conducted. The *Anchor* concepts correspond to a given individual who is slightly moving within the scene. The *Studio* concepts correspond to a given background scene. Concepts *Basketball* and *Hockey* correspond to a sport action which is usually characterized by some camera motions and specific feature trajectories. Concepts *Car* and *Map* correspond to a given rigid objects.

Figure 3 presents the mean Precision/Recall obtained through the simulations for all concepts. Each curve represents a relevance feedback iteration. It shows

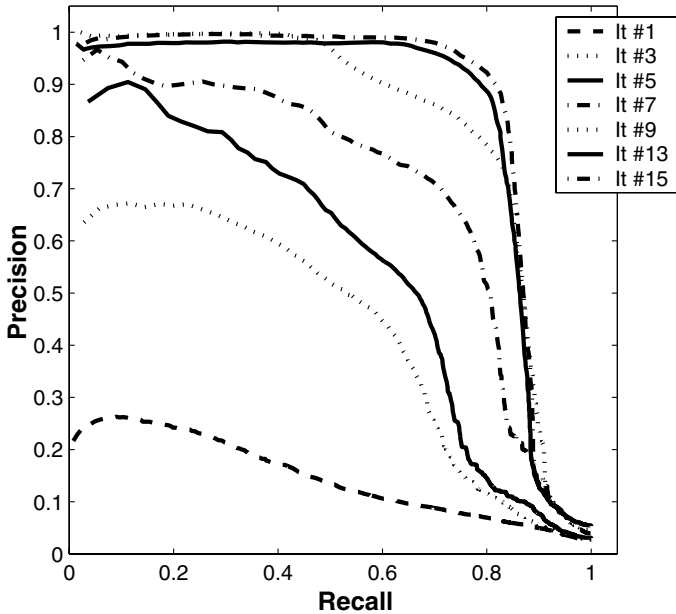


Fig. 3. Mean Precision/Recall curves obtained for every relevance feedback iteration

that the Local Feature Dynamics Model (LFDM) we propose, is well-suited to capture the visual content of video sequences. Furthermore, the increase of the retrieval accuracy through the relevance feedback iterations shows that the algorithm is able to select within positive examples the relevant local feature dynamics that define the spatio-temporal pattern queried by the user.

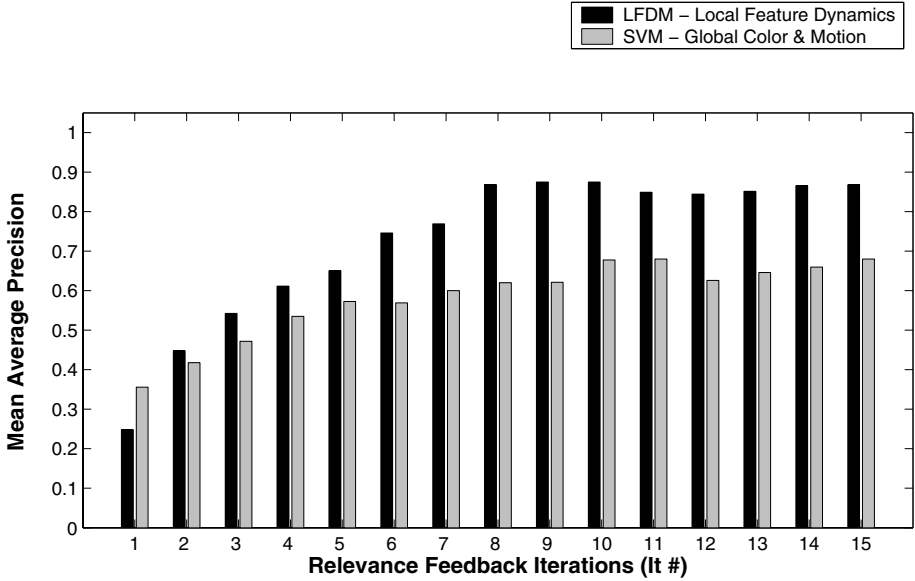


Fig. 4. Comparison of the MAP for every relevance feedback iteration

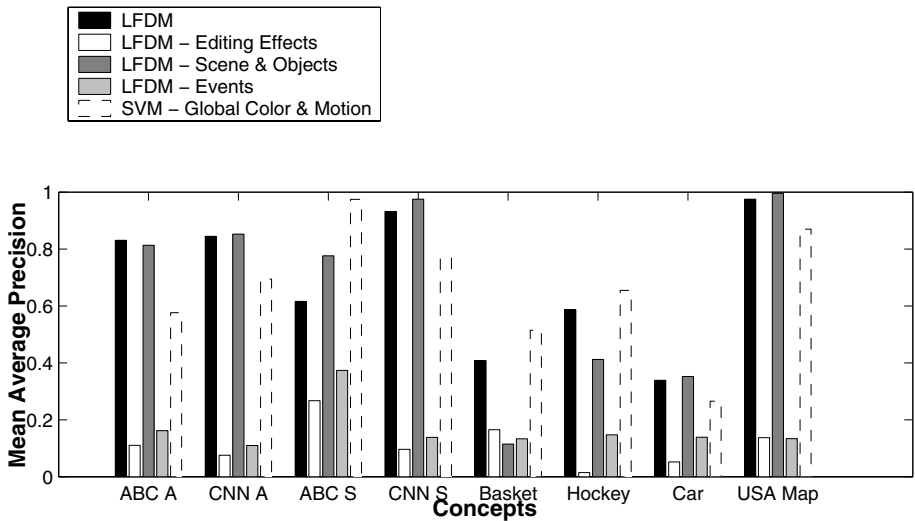


Fig. 5. Detailed MAP obtained for the last relevance feedback iteration (It #15)

Figure 4 presents the Mean Average Precision (MAP) obtained for every relevance feedback iteration, comparing the LFDM with the approach presented in [2] which uses a Support Vector Machine (SVM) to learn the concept from global color and motion histograms (MPEG motion vectors) with the positive and the negative examples provided by the user. The LFDM approach needs more example to be able to learn the concept. However it is able to converge to the concept while the global approach tends to stop learning after some iterations.

Finally, figure 5 details the performances obtained for every concept. It presents the MAP obtained with the LFDM using all information, the LFDM using only the camera motions (LFDM - *Editing Effects*), the LFDM using only the local feature appearances (LFDM *Scene Objects*), the LFDM using only the local feature dynamics (LFDM - *Events*) and for comparison purposes, the global SVM approach. It is clear that appearance model (LFDM - Scene Objects) is the main component of the LFDM. It even performs slightly better for the *Studio* and for the *Map* concepts. For these concepts, the information provided by the camera motions and the local feature trajectories is useless and even tends to degrade the performance of the whole model. However, for concepts involving dynamics, our approach successfully exploits the different information about the appearances and the dynamics of the scene.

6 Conclusion

In this paper, a method to index the visual content of video sequences based on local feature dynamics has been presented. Based on this index, a model of visual queries has been proposed that is estimated online, adaptively, through the user interactions captured within a relevance feedback loop. Using this method, arbitrary visual parts of video sequences may be queried, under the assumption that there is enough examples within the collection to accurately estimate the query.

We believe that this approach provides a way to perform efficient and reliable content-based retrieval of video sequences. The solution does not make any assumption about the content, is scalable and performs well in terms of response time and precision/recall.

Future works will consider more sophisticated models of visual queries, considering the relative scale and position of local features along the temporal dimension and the associated index structures that would permit such models to be efficiently used.

References

1. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
2. Eric Bruno, Nicolas Moënne-Loccoz, and Stéphane Marchand-Maillet. Learning user queries in multimodal dissimilarity spaces. In *Third International Workshop on Adaptive Multimedia AMR'05*, July 2005.

3. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, june 2003.
4. M.A. Fisher and R.C: Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, pages 381–395, 1981.
5. W. Förstner. A feature-based correspondence algorithm for image matching. In *Int. Arch. Photogrammetry and Remote Sensing*, volume 26, page 150166, 1986.
6. C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 189–192, 1988.
7. Bruno Janvier, Eric Bruno, Stéphane Marchand-Maillet, and Thierry Pun. Information-theoretic framework for the joint temporal partitioning and representation of video data. In *Proceedings of the European Conference on Content-based Multimedia Indexing, CBMI'03*, September 2003.
8. T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. sp, may 2004.
9. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
10. Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Learning local affine-invariant part models for object class recognition. In *Workshop on Learning, Snowbird, Utah*, 2004.
11. F. Li, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Ninth IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 1134, 2003.
12. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
13. David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
14. Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *8th International Conference on Computer Vision*, pages 525–531, 2001.
15. Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002. Copenhagen.
16. Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
17. Nicolas Moëgne-Loccoz, Bruno Janvier, Stéphane Marchand-Maillet, and Eric Bruno. Managing video collections at large. In *Proceedings of the First Workshop on Computer Vision Meets Databases CVDB'04*, Paris, France, 2004.
18. F. Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Segmenting, modeling and matching video clips containing multiple moving objects. In *IEEE Conference on Computer Vision*, volume 2, pages 914–921, 2004.
19. Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.
20. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, October 2003.

21. Q. Tian, N. Sebe, M. S. Lew, E. Louprias, and T. S. Huang. Image retrieval using wavelet-based salient points. In *Journal of Electronic Imaging, Special Issue on Storage and Retrieval of Digital Media*, pages 835–849, 2001.
22. Philip H. S. Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *Workshop on Vision Algorithms*, pages 278–294, 1999.
23. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, 1996.

Temporal Relation Analysis in Audiovisual Documents for Complementary Descriptive Information

Zein Al Abidin Ibrahim, Isabelle Ferrane, and Philippe Joly

University Paul Sabatier, IRIT, Toulouse
{Ibrahim, Ferrane, Joly}@irit.fr

Abstract. Relations among temporal intervals can be used to detect semantic events in audio visual documents. The aim of our work is to study all the relations that can be observed between different segmentations of a same document. These segmentations are automatically provided by a set of tools. Each tool determines temporal units according to specific low or mid-level features. All this work is achieved without any prior information about the document type (sport, news ...), its structure, or the type of the semantic events we are looking for. Considering binary temporal relations between each couple of segmentations, a parametric representation is proposed. Using this representation, observations are made about temporal relation frequencies. When using relevant segmentations, some semantic events can be inferred from these observations. If they are not relevant enough, or if we are looking for semantic events of a higher level, conjunctions between two temporal relations can turn to be more efficient. In order to illustrate how observations can be made in the parametric representation, an example is given using Allen's relations. Finally, we present some first results of an experimental phase made on TV news programs.

1 Introduction

Many automatic tools for audiovisual content indexing produce results as segments or temporal portions identifying the presence or not of a specific object. Most of them are based on several low level features extracted in a previous step (dominant color, applauses or speech on the soundtrack, text or character presence on the screen, movement quantity ...). Their main purpose is to detect relevant events, in order to generate summaries or to determine temporal entries in the video stream. Then, these events can either be used as indexes or as input for further processing steps. Indexes produced by this way generally suffer from lack of semantics. This is partly due to the usage of only one modality in feature extraction. Combining primary indexes that characterize an audiovisual content can improve the semantic level of outputs and produce more meaningful indexes.

Indexing tools can be classified on the base of features involved in the analysis process. In the literature, features extracted from motion analysis are used in [1] to detect the complete set of events that can be find in a video of soccer game, and in [2] to extract highlights and events from the same type of video. Some other techniques are based on colour and movement features as in [3] to classify the soccer video game in two phases: play and break. Colour and shape are used in [4] to classify TV news

in semantic classes. Classification of basketball videos in semantic classes using some rules is made in [5] by a combination of colour, shape, movement, and texture. Other techniques based on audio features like in [6] aim at extracting the highlights in baseball videos, or classifying events of football game audio in semantic classes as in [7]. Multimodal features are also used in [8] to index the news videos by detecting six semantic classes, in [9] to detect the highlights of baseball videos and in [10] to extract the highlights in TV programs of formula 1.

All these techniques are specialised because they generally depend on document types, event types or production rules. Their need of using prior knowledge can be consider as a limitation from a generality point of view. They are useless, or need to be adapted if there is a change in the use context, that is to say if they are applied to a new type of content or used for retrieving some events which are not predefined. Although some efforts for generalizing event detection techniques have been made, for example, in [11] where generalisation is made at sports video level, they still remain limited to a specific domain.

In order to avoid this limitation problem, we have focused on finding techniques that are independent from any prior knowledge.

The first step of our work is to consider the results of different segmentation systems, which can be applied on the different media (audio, image, image sequence, text ...) of a same multimedia document. These results provide information on the content evolution at a low and mid semantic level. Results of a given couple of segmentations are analyzed in order to observe temporal relations between events. Some complementary descriptive information on the document temporal structure can then be deduced from these observations. Then we propose to study conjunction of temporal relations as a mean of extracting higher complementary information.

This paper will be organized as follows: in section two, we will study research work carried out in the domain of temporal information. Then, we will present our parametric representation of temporal relation between two segments. Analyzing each temporal relation that can exist between results of a couple of segmentation will lead us to propose a data structure to memorize each observation. Using this structure to identify relevant relations will need to address inherent quantization and classification problems. From the proposed parametric representation, relations will be represented in a graphical way and conjunction between temporal relations will be studied. At the end of the second section, we will present an example of our approach by applying it to Allen's relations. A discussion of neighboring relations for error handling will be presented in section three. In section four, we will give some results of experimental works carried out on a set of four temporal segmentations of a TV news program. At last, in section five, we will conclude and present our future works on this topic.

2 Temporal Representation

2.1 Related Work

Temporal representation is an essential topic for any activity studying changes along the time dimension. This is why many disciplines are addressing this important problem. Because an event can be produced by interaction of multimedia objects,

analysing temporal relations between events in an audiovisual document is an important issue for content indexing. Results of such an analysis can be used to match up a given content with a predefined temporal structure (by means of hierarchical HMMs for example) in order to identify specific highlights, or to automatically build a temporal representation of the content evolution. The state of the art demonstrates that such tools are always built on a priori knowledge on how events are temporally related to each other in audiovisual documents. For example, we can use the fact that anchor frames alternate with reports in a TV news program. In the same way, we can take into account the fact that songs are followed by applause on entertainment program soundtracks, or that a goal in a soccer game, could have been marked when the ball crosses the goal region and when an explosion of audience's voices follows immediately.

To extend the temporal analysis of video or audio contents to subtle or unpredictable relations between any kind of events, tools to represent and to reason about time must be involved in the process.

Basics of the representation of time have been given by Hayes who introduced six notions of time to represent temporal relations in [12], that is to say: the basic physical dimension, the time-line, time intervals, time points, amount of time or duration, and time positions. This problem has been addressed by several researchers. We can find overviews of different approaches to temporal representation and reasoning in a survey by Chittaro and Montanari ([13] and [14]), in a survey by Vila ([15]), and in an article by Pani and al. ([16]).

Existing models to express temporal relationships can be divided into two classes: point-based ([17]) and interval-based models ([18]).

In point-based models, points are elementary units along the time dimension. Each event is associated with a time point. Given two events e_1 and e_2 , three temporal relations can be determined between them. An event can be *before* ($<$), *after* ($>$) or *simultaneous* to ($=$) a second event. These relations are called the basic point relations.

In some cases, relations between events may be indefinite. For example, we know that an event e_1 can not occur after an event e_2 . This means that e_1 is either *before* or *simultaneous* to e_2 and can be represented by a disjunction of basic point's relations like $e_1 \{<, =\} e_2$. Since there are 3 basic relations, $2^3 = 8$ disjunctions exist, each one representing an indefinite relation.

An example of a point-based representation is the timeline, on which media objects are placed on several time axes. Though this representation is also used as an interval-based representation, we can find the timeline model applied in various applications such as HyTime ([19]).

Interval-based models consider elementary media entities as time intervals which can be ordered according to different relations. The existing models are mainly based on the relations defined by Allen in [18] to express the knowledge about time. The interval can be seen as a point that has duration along the time dimension.

2.2 Parametric Representation of Temporal Relations

Let us consider that two temporal segmentations S_1 and S_2 of a same video document are performed in order to use their results in an analysis processing. This first step

produces segments seen as temporal intervals localizing portions of the video where the presence of a specific feature is detected. Each segmentation system identifies a sequence of intervals where one type of event occurs (segments of gradual transition effects, all appearances of a given character, moments where some music can be heard on the soundtrack, etc). Hence, a segmentation corresponds to a set of temporally disjointed segments: $S1 = \{s_{1i}\}_{i \in [1, N]}$ and $S2 = \{s_{2j}\}_{j \in [1, M]}$

A temporal interval is characterised by its two endpoints, or instants. Let s_{1i} and s_{2j} be two segments characterised by their beginning and their ending points, respectively $[s_{1ib}, s_{1ie}]$ and $[s_{2jb}, s_{2je}]$ on each segmentation. We can represent the temporal relation between these segments with three variables, as proposed in [20]:

- 1) **Lap** = $s_{2jb} - s_{1ie}$
- 2) **DB** = $s_{1ib} - s_{2jb}$
- 3) **DE** = $s_{2je} - s_{1ie}$

This representation will be used in our first step to calculate a matrix that will contain the occurrences of each relation between events and in the next step to calculate the conjunction of two relations. This parametric representation can be used in the point based model where events have no duration. In this case, we will have:

- 1) Lap = $e2 - e1$
- 2) DB = $e1 - e2$
- 3) DE = $e2 - e1$

The values of the parameters are quasi-equal. Only one parameter can be used to represent a relation in the point-based model (Lap = DE = -DB).

2.2.1 TRM Calculation

A temporal relation between two segments can be represented using the three parameters mentioned above. Thus, from a graphical point of view, a relation between two intervals will be modeled as a 3D point. For two segmentations **S1** and **S2**, the three parameters between each couple of segments (s_{1i}, s_{2j}) can be evaluated and so represented by a point in the corresponding 3D space. The relation between s_{1i} and s_{2j} can be seen as:

$$s_{1i} \mathbf{R}(DE, DB, Lap) s_{2j}$$

For each 3D point (ie. for each potential temporal relation between two segments), we associate a vote accumulator that counts the numbers of times this relation is observed between both segmentations. Then we obtain a matrix of accumulators called the Temporal Relation Matrix (TRM). It can be used directly to determine the frequencies of potential relations. It can also be used to observe remarkable distributions of votes and so to identify a general rule about the temporal behavior of events. For each occurrence of a given relation **R** represented by the value of the parameter vector), a vote is added to the associated accumulator in the TRM. For example, the relation

s_{1i} $R(DE, DB, Lap)$ s_{2j} will correspond to the cell $TRM[DE][DB][Lap]$.

If we consider now occurrences of a temporal relation R' that can be observed between s_{2j} and s_{1i} (s_{2j} $R'(DE', DB', Lap')$ s_{1i}), we can estimate its parameters using those of the relation R between s_{1i} and s_{2j} . We can so establish that

$$\begin{aligned} DE' &= -DE \\ DB' &= -DB \\ Lap' &= -DE + DB + Lap \end{aligned}$$

So the TRM associated with the relations that can be observed between two segmentations $S2$ and $S1$ ($TMR(S2, S1)$) can be calculated using the TRM ($S1, S2$) by:

$$TRM(S2, S1)[i][j][k] = TRM(S1, S2)[-i][-j][-i + j + k]$$

The size of the TRM, is directly related with the document duration, and so may be very large. Thus, the first problem to overcome is the quantization of the 3D space in order to generate a matrix of an acceptable size. Once the matrix is created, initialized and filled in with the votes, the analysis step can be performed.

Quantization

The quantization step of the matrix depends on the scale of the low level features. For audiovisual contents, low features may be associated to one value for each frame or integrated per windows of 1 second. Audio analysis may produce results, which do not necessarily correspond to any video frame time stamp. Therefore, for our observations, we have to quantify each dimension to define intervals based on the largest temporal scale used to express the features.

In a more general manner, quantifying this space leads to the same problems that are generally identified for vote methods. Among them, the size of the matrix has to be limited. Since the maximal variation of the parameters is less or equal to the difference between the end of the last interval and the beginning of the first one in the two analyzed sequences, we can a priori identify the maximal boundaries of this space. Furthermore, in the case of a space of high dimension, we can directly apply a hierarchical discretization process when starting with coarse quantization and progressively focusing only on subparts of the space which are receiving more votes than the other [21].

2.2.2 Conjunction Problem

The conjunction of temporal relations is tackled by many researchers working on Allen's temporal relations. Their purpose is to calculate possible relations that can result from a conjunction of relations. In this case, conjunction is considered from a qualitative point of view. In our case, and since we are representing relations by means of parameters that express constraints on the ending points of two intervals, we can consider conjunction from a quantitative point of view because we can check that the result of the conjunction of two relations is characterized by the same three parameters.

Let us consider three temporal intervals s_{1i} , s_{2j} , and s_{3k} and two relations between these intervals:

$$s_{1i} R1(a1,b1,c1) s_{2j}, \text{ and } s_{2j} R2(a2,b2,c2) s_{3k}.$$

The relation called R3 that results from the conjunction of these two relations R1 and R2 can be defined as well by three parameters. These parameters are calculated with the parameters of the two relations R1 and R2:

$$s_{1i} R3 (a1+a2,b1+b2,c1-b2) s_{3k}$$

2.2.3 Classification

For each couple of available segmentations, a TRM is generated. As it has been explain in section 2.2.1., this TRM will contain the votes of all possible relations between each segment of the first segmentation and each segment of the second one. An analysis step has to be performed in order to search for the most significant relations between each couple of features. Unlike other vote techniques, the significant relation can not only have a maximum value in a cell but actually, most of semantic temporal relations determine subparts of the TRM where votes are distributed. So, the first step of the TRM analysis is to localize zones in the 3D space regarding to the vote distribution. This localization can be achieved by clustering methods or classification systems. Another approach consists in subdividing the TRM into parts according to prior knowledge about semantic relations like for example the Allen's relations. This approach consists in associating each semantic relation with a subpart of the vote space (in the TRM) defined by the constraints that govern ending points. In this case, the occurrence of the relation **R** between two features is computed as the sum of the votes contained in the related subpart. The size of the TRM will be reduced to a cubic 3D matrix whose dimensions will be equal to the number of predefined relations.

2.2.4 Graphical Representation of Predefined Relations

The most fundamental and well-known theory about reasoning with time intervals is formulated as the Allen's algebra. This algebra describes possible relations between 1-D intervals. Allen proposed a complete set of relations between two intervals. For two given intervals, there are 13 distinct possibilities to temporally relate these segments. These 13 relations (Table 1) can be represented by 6x2 zones (corresponding to direct and inverse relations) and a last one which corresponds to the fact that two segments may have the same beginning and same ending points.

By analogy with the point relations, $2^{13} = 8192$ indefinite interval relations can be defined as disjunctions of the basic interval relations. Since some temporal models are point-based and some others are interval-based, a switch between the models is made by representing the interval relations as conjunctions of point basic relations between the segment boundaries (Table 1).

Since events that are far in distance may not provide important information about the content or the structure of the document, we will limit the detection of the 'before' and 'after' relations by a parameter ' α ' that expresses the maximum distance that can separate two intervals.

Table 1. Constraints of Allen relations

Relation	Symbol and Inverse	Point Notation	Example
s_{1i} before(α) s_{2j}	< >	$s_{1ib} < s_{1ie} < s_{2jb} < s_{2je}$ & $(0 < s_{2jb} - s_{1ie} \leq \alpha)$	AAA $\begin{matrix} \leftarrow \\ \rightarrow \end{matrix}$ BBBB $d \leq \alpha$
s_{1i} meets s_{2j}	m mi	$s_{1ib} < s_{1ie} = s_{2jb} < s_{2je}$	AAAAA BBBBB
s_{1i} overlaps s_{2j}	o oi	$s_{1ib} < s_{2jb} < s_{1ie} < s_{2je}$	AAAAA BBBBBB
s_{1i} starts s_{2j}	s si	$s_{1ib} = s_{2jb} < s_{1ie} < s_{2je}$	AAAA BBBBBBBB
s_{1i} finishes s_{2j}	f fi	$s_{2jb} < s_{1ib} < s_{1ie} = s_{2je}$	AAA BBBBBBBB
s_{1i} equals s_{2j}	= =	$s_{1ib} = s_{2jb} < s_{1ie} = s_{2je}$	AAAAAA BBBBBB
s_{1i} during s_{2j}	d di	$s_{2jb} < s_{1ib} < s_{1ie} < s_{2je}$	AAAAA BBBBBBBBB

If we represent the temporal relation between two intervals with the three parameters **Lap**, **DE**, **DB**, then Allen's relations lay down some constraints on these parameters as shown in Table 2.

For example, the 'during' relation corresponds to the following definition:

$$s_{2jb} < s_{1ib} < s_{1ie} < s_{2je}$$

In this case, $s_{2jb} - s_{1ie} < s_{1ib} - s_{1ie} < s_{1ie} - s_{1ie} < s_{2je} - s_{1ie}$

$$\Rightarrow \mathbf{Lap} < 0 < \mathbf{DE}.$$

And $s_{2jb} - s_{2jb} < s_{1ib} - s_{2jb} < s_{1ie} - s_{2jb} < s_{2je} - s_{2jb}$

$$\Rightarrow 0 < \mathbf{DB} < -\mathbf{Lap}$$

For the *mi* relation, we have the following constraints:

$$s_{2jb} < s_{2je} = s_{1ib} < s_{1ie}$$

$$s_{2jb} - s_{2jb} < s_{1ib} - s_{2jb} = s_{2je} - s_{2jb} < s_{1ie} - s_{2jb} \Rightarrow 0 < \mathbf{DB}.$$

$$s_{2jb} - s_{1ie} < s_{2je} - s_{1ie} < s_{1ie} - s_{1ie} \Rightarrow \mathbf{DE} < 0.$$

$$s_{2jb} - s_{1ie} < s_{2je} - s_{1ie} = s_{1ib} - s_{1ie} < 0 \Rightarrow \mathbf{Lap} < 0$$

$\mathbf{DE} - \mathbf{DB} = s_{2je} - s_{1ie} - s_{1ib} + s_{2jb} = (s_{2jb} - s_{1ie}) + (s_{2je} - s_{1ib}) = \mathbf{Lap}$, $s_{2je} - s_{1ib} = 0$ because s_{2j} 'meets' s_{1i} .

By the same way, we can derive the constraints of the 13 Allen's relations between intervals.

These constraints define subparts in the TRM identifying so the place where the votes of each relation go.

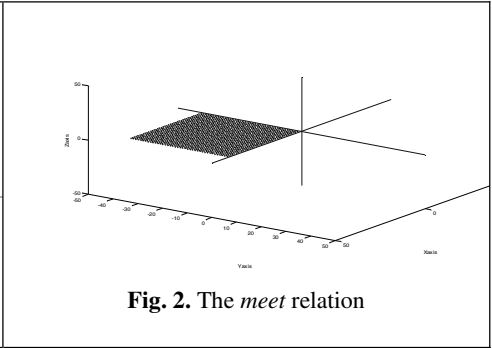
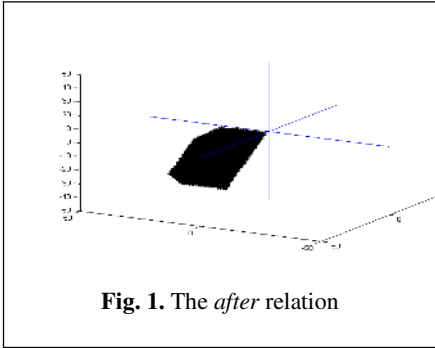
An a priori quantization of the space using Allen relations is defined following the rules given in Table 2.

For abbreviation reasons, we will note $\mathbf{DE}=x$, $\mathbf{DB}=y$, and $\mathbf{LAP}=z$;

Let us consider the region defined by the *after* relation. Table 2 indicates that the relation between two intervals s_{1i} and s_{2j} is an *after* relation iff:

Table 2. Constraints of Allen relations in (DE,DB,Lap) space

<i>Relation</i>	<i>Lap</i>	<i>DB</i>	<i>DE</i>
<	$0 < Lap \leq \alpha$	$DB < -Lap < 0$	$DE > Lap > 0$
<i>m</i>	$Lap = 0$	$DB < 0$	$DE > 0$
<i>o</i>	$Lap < 0$	$DB < 0$	$DE > 0$
<i>s</i>	$Lap < 0$	$DB = 0$	$DE > 0$
<i>f</i>	$Lap < 0$	$DB > 0$	$DE = 0$
=	$Lap < 0$	$DB = 0$	$DE = 0$
<i>d</i>	$Lap < 0$	$DB < -Lap$	$DE > 0$
>	$DE - DB < Lap < 0$ & $0 < DB - DE + Lap \leq \alpha$	$DB > 0$	$DE < 0$
<i>mi</i>	$Lap = DE - DB$	$DB > 0$	$DE < 0$
<i>oi</i>	$Lap < DE - DB < 0$	$DB > 0$	$DE < 0$
<i>si</i>	$Lap < DE$	$DB = 0$	$DE < 0$
<i>fi</i>	$Lap < 0$	$DB < 0$	$DE = 0$
<i>di</i>	$Lap < DE$	$DB < 0$	$DE < 0$



- 1) $x - y < z < 0$ & $y - x + z \leq \alpha$
- 2) $y > 0$
- 3) $x < 0$

So, votes for the *after* relation (Fig. 1) will go in a zone delimited by the constraints mentioned above.

The *meet* relation (Fig. 2) corresponds to a restricted area of the plane defined by the equation: $z = 0$;

More precisely, s_{1i} and s_{2j} verify the *meet* relation iff:

- 1) $z = 0$
- 2) $y < 0$
- 3) $x > 0$

The equal relation (Fig. 3) corresponds to the 3D line traced in the graph.

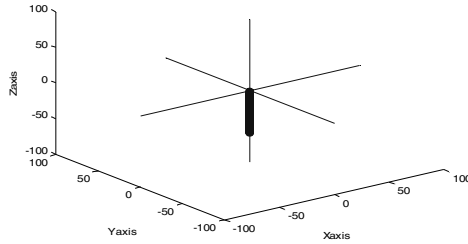


Fig. 3. The *Equal* relation

$A \mathbf{R} B$ and $\mathbf{R} = \text{'equal'}$ iff

- 1) $z < 0$
- 2) $y = 0$
- 3) $x = 0$

By the same way, we can identify the TRM regions corresponding to all the other relations.

With this representation, we can check to which pre-identified relation a given observation between two intervals belongs to. In other words, the observation parameters shall verify the equations defining a zone to identify the corresponding semantic relation.

2.3 Neighboring Relations for Error Handling

The segments provided by an automatic segmentation process may be imprecise. The sources of imprecision may originate from the poor quality of the low level feature extraction from the video streams. It can be due to the quantization problem discussed in the previous section. The wrong extraction of segments may lead to vote in a wrong cell, that is to say that a vote added in the TRM will strengthen an unobserved relation instead of the relation that truly exists and should have been reinforced by this vote. In this case, we have to deal with errors and to try to decrease their effects on the analysis process. To do so, a report of votes in the neighbourhood of a relation can be performed.

In the case of Allen's relation and as most of the approaches in this domain, we can use the neighbouring principle of Freska ([22]). It can be seen as: let A and B be two events verifying the temporal relation '*meet*'. Then by moving or slightly deforming the objects, we can change the relation to '*before*' or '*overlap*'. Therefore, '*before*' and '*overlap*' are conceptual neighbours of '*meet*'. The relation '*equal*', for example, is not a conceptual neighbour of '*meet*', because '*equal*' can not be obtained directly from '*meet*' by deforming or temporally moving events. In our case, each relation is represented by a zone and its neighbours are the zones that are close to it in this space.

A topological extension of the Freska's neighbourhood principle to any relation can be formally defined by:

Let $R_i (X_i, Y_i, Z_i)$ and $R_j (X_j, Y_j, Z_j)$ be two compact zones in the TRM and let $R_k = R_i \cap R_j$ where $X_k = (X_i \cap X_j)$, $Y_k = (Y_i \cap Y_j)$, and $Z_k = (Z_i \cap Z_j)$.

R_i has R_j as direct neighbour if one and only one of the R_k parameters is empty.

For example, let us consider $R_i = \mathbf{Meet}$ and $R_j = \mathbf{Overlap}$. In this case, X_i corresponds to $\mathbf{DE} > 0$, Y_i to $\mathbf{DB} < 0$, and Z_i to $\mathbf{Lap} = 0$.

$$R_i (X_i, Y_i, Z_i) = \mathbf{M} (]0 +\infty[,]-\infty 0[, \{0\})$$

$$R_j (X_j, Y_j, Z_j) = \mathbf{O} (]0 +\infty[,]-\infty 0[,]-\infty 0[)$$

We have $(X_i \cap X_j) =]0 +\infty[$, $(Y_i \cap Y_j) =]-\infty 0[$, and $(Z_i \cap Z_j) = \emptyset$. As only $(Z_i \cap Z_j) = \emptyset$, then \mathbf{M} and \mathbf{O} are direct neighbours.

This is not the case for \mathbf{Meet} and \mathbf{During} where both $Y_k = \emptyset$ and $Z_k = \emptyset$.

Figure 4 shows three neighbour relations: Meet (intermediate grey), Overlap (light grey), and start (dark grey).

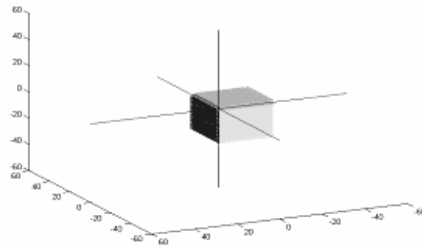


Fig. 4. Neighbor relations: Meet, Overlap and Start

2.4 Impact of Recall/Precision on the TRM

Imprecision of automatic segmentation process can also bring us to the question of tool performances and their impact on the TRM. Recall and precision are basic measures that are used as performance criteria for segmentation tool evaluation. The former is the ratio of the number of segments rightly detected (N_r) to the total number of segments in the reference segmentation (N_{ref}). The latter is the ratio of the number of segments rightly detected (N_r) to the total number of segments rightly or wrongly detected ($N_r + N_w$). $\text{Recall} = N_r / N_{tot}$ $\text{Precision} = N_r / (N_r + N_w)$

Considering these measures, we can study their effect on the votes and the TRM.

Let S_1 and S_2 be two segment sequences of the same document, generated by two different segmentation processes (Pr_1 and Pr_2 respectively) each one based on specific features. Let R_1 and P_1 (R_2 and P_2 respectively) be the recall and the precision of the segmentation process Pr_1 (Pr_2). The impact of R_1 , P_1 , R_2 , and P_2 on the TRM can be expressed by:

If $R_1 \ll P_1$ ($R_2 \ll P_2$), this will mean that there are missing intervals ($N_r + N_w \ll N_{tot}$). In this case, the votes in the TRM will decrease, which will also lead the recall of the TRM to decrease.

On the other hand, if $P_1 \ll R_1$ ($P_2 \ll R_2$), this will mean that too many segments have been detected ($N_r + N_w \gg N_{tot}$). In this case, the votes in the TRM will increase, which will lead the precision of the TRM to decrease.

3 Experimental Observations

We have computed several TRMs on temporal segmentations of TV news programs. Five different segmentations have been involved, each one related to the presence or not of a specific feature: (1) Newscaster on the screen, (2) speech (3) silence (4) music and (5) applause on the soundtrack.

We have built the TRMs to observe temporal relations between the newscaster and each audio event: newscaster and speech, newscaster and silence, newscaster and music, and finally newscaster and applaudes.

Generally, votes are distributed along parallel planes that have the following equation: $y + z + d = 0$, where d is duration of the intervals of the first segmentation. By the same way, we can observe the distribution of the points in parallel planes that have the equation: $x - z - d = 0$, where d is the duration of the intervals of the second segmentation. The switch between the planes that corresponds to intervals of different durations is done by a translation of a distance that is equal to the difference between the durations.

Particularly, we observe on figure 5 that points representing relations between the newscaster and speech segments are distributed along parallel lines. The circles represent the points excluded after the quantification step and considered as meaningless (ie. when $\text{lap} > \alpha$). After the TRM projection on the plane XOZ (Fig.6), we observe that the points make a thick line which passes through the origin of that plane.

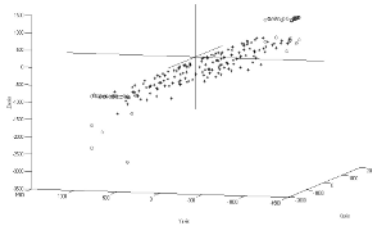


Fig. 5. Relations between newscaster and speech segments

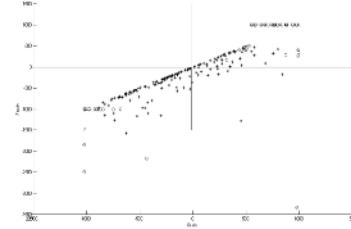


Fig. 6. Projection of fig. 4 on the plane XOZ

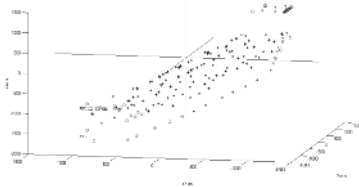


Fig. 7. Relations between newscaster and music segments

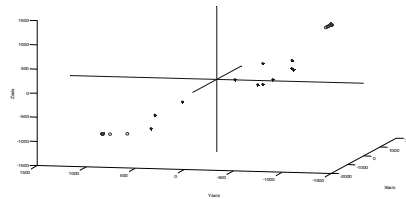


Fig. 8. Relations between newscaster and applaudes segments

We see that the distribution of most of the points are parallel lines, all included in the plane of equation $z = ay + b$; for x arbitrary. This line distribution can be seen in Fig 5 and 9.

We observe that the points that represent the relations (Newscaster, music) (Fig. 7), and (Newscaster, applause) (Fig. 8) which show the case that when the newscaster talks, we hear neither music, nor applause, are very rare on the figures. We can observe that there are much points related to (newscaster, music) since the news video contain publicity segments where we can find music but not applause segments. For the relation between (Newscaster, silence) (Fig. 9), the number of points increases and the projection of the points on the XOZ plane (Fig. 10) is a straight line crossing the origin.

On figure 10, we can observe that votes are distributed along a plane where $x \approx z$ It means that actually $DE \approx LAP$, and so that $s_{2je} - s_{1ie} \approx s_{2jb} - s_{1ie}$. We deduce that $s_{2je} \approx s_{2jb}$ which means that segments of the second features are rather short. The second feature is corresponding to silent segments in a TV News program where they are actually really short.

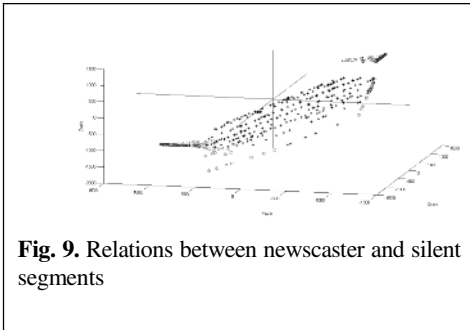


Fig. 9. Relations between newscaster and silent segments

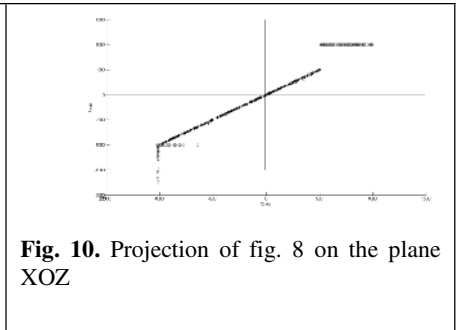


Fig. 10. Projection of fig. 8 on the plane XOZ

4 Conclusion and Future Works

In this paper, we have presented a new technique to highlight significant temporal relationships between events segmented by automatic analysis tools. Relations between two temporal intervals are represented by a 3D point. After representing all the relations that can be observed between two temporal segmentations, a classification of points can be achieved. The observation space must be discretized in zones, each one associated with a semantic relation. This classification of the points can be also achieved on combined TRMs using the conjunction formula. Then we discussed how to use the Allen’s relations to give an example of the discretization process. At last, we presented results obtained on a news video program and observed the obtained distribution of the points in the space.

Different potential extensions can be considered for this work. First, problems related with imperfection of the extracted temporal information have to be taken into account. As it has been mentioned before, votes in the TRM may be distributed in the neighborhood of the corresponding relation. The calculation of the neighbor relations in the TRM space may be done using the distance between each point to the closest

observed 3d zones. Then, votes may be distributed as weights following a function of the inverse value of the distance. A part of the TRM associated to a relation class is a closer neighbor to another one when this distance is smaller. So, a neighbor tree can be automatically built for each given partition of the TRM. Furthermore, weights associated to a vote can be distributed in a shaper way while using the TRM topology instead of using a neighbor tree. The weight computation may depend on the position of the point in a zone (i.e. on the distance between this point and the center of the zone), the distance that separates it from other zones, the zone size, and the number of included points.

We also intend to explore the conjunction of observed relations in a hierarchical way. Being able to handle a conjunction of a large set of relations should allow us identify complex temporal events.

Working on video document content and trying to observe and detect relevant temporal relations could be used for reasoning purposes. The MPEG7 formalism offers a mechanism for temporal decomposition of audio visual documents and segmentation description. On the other hand, a logical formalism can express relations between concepts through a set of logical rules. Considering segments as basic concepts and applying some inference rules could be a way to deduce some high-level knowledge about the document content. This approach could be studied in future works as well as the way to explore the usage of the TRM as a style signature which may characterize the temporal evolution of different types of documents.

References

1. V. Tovinkere, R. J. Qian, "Detecting Semantic Events in Soccer Games: Toward a Complete Solution", Proc. ICME'2001, pp. 1040-1043, Aug. 2001, Tokyo, Japan.
2. A. Bonzanini, R. Leonardi, P. Migliorati, "Exploitation of Temporal Dependencies of Descriptors to Extract Semantic information", DEA - University of Brescia, Via Branze, 38 - 25123, Brescia, Italy.
3. L. Xie, S-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with Hidden Markov Models," Proc. IEEE Int'l. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2002.
4. Y. Avrithis, N. Tsapatsoulis and S. Kollias. "Broadcast News Parsing Using Visual Cues: A Robust Face Detection Approach". IEEE International Conference on Multimedia and Expo, New York City, NY, July 2000.
5. W. Zhou, A. Vellaikal, C.-C. J. Kuo, Rule-based Video Classification System for Basketball Video Indexing, in ACM Mult. Conf., 2000.
6. Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in Proc. ACM Multimedia 2002, pp. 105-115, 2000, Los Angeles, CA, USA.
7. S. Lefevre, B. Maillard, and N. Vincent, "3 classes segmentation for analysis of football audio sequences," in Proc. IC DSP'2002, July 2002, Santorin, Greece.
8. S. Eickeler and S. Muller, "Content-Based Video Indexing of TV Broadcast News Using Hidden Markov Models". In Proc. IEEE ICASSP, Phoenix, 1999.
9. M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method," in Proc. ACM Multimedia 2002, Dec. 2002, Juan Les Pins, France.
10. M. Petrovic, V. Mihajlovic, W. Jonker, and S. Djordjevic-Kajan, "Multi-modal extraction of highlights from tv formula 1 programs," in Proc. ICME'2002, Aug. 2002, Lausanne, Switzerland.

11. L. Duan , M. Xu , Xiao-Dong Yu , Qi Tian. "A unified framework for semantic shot classification in sports videos," Proceedings of the tenth ACM international conference on Multimedia, December 01-06, 2002, Juan-les-Pins, France.
12. Hayes, Patrick. 1996. "A Catalog of temporal theories." Technical report UIUC-BI-AI-96-01, University of Illinois 1995.
13. L. Chittaro and A. Montanari, "Trends in Temporal Representation and Reasoning". The Knowledge Engineering Review 11(3): 281-288, 1996.
14. L. Chittaro and A. Montanari, "Temporal Representation and Reasoning in Artificial Intelligence: Issues and Approaches", Annals of Mathematics and Artificial Intelligence, Vol.28, 2000, pp. 47-106.
15. L. Vila, "A Survey on Temporal Reasoning in Artificial Intelligence", Artificial Intelligence Communications, Vol. 7(1), 1994, pp. 4-28.
16. A. K. Pani. Temporal representation and reasoning in artificial intelligence: A review. *Mathematical and Computer Modelling*, 34:55–80, 2001.
17. M. Vilain, and H. A. Kautz, "Constraint propagation algorithms for temporal reasoning", In AAAI-86, 1986, pp. 132-144.
18. J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communication of ACM*, 26(11):832 – 843, 1983.
19. HyTime Information Technology, "Hypermedia / Time-based Structuring Language (HyTime)", ISO/IEC 10743, novembre 1992.
20. B. Moulin. "Conceptual graph approach for the representation of temporal information in discourse," Knowledge based systems, vol. 5, num. 3, 1992, pp 183 –192.
21. H. Li and M. A. Lavin . "Fast Hough Transform: A Hierarchical Approach", *Journal on Graphical Models and Image Processing (CVGIP)* (36), 139-161, Novembre/Décembre 1986.
22. C. Freska, "Temporal Reasoning Based on Semi-intervals", *Artificial Intelligence*, 54:199-227 (1992).

Using Segmented Objects in Ostensive Video Shot Retrieval

Sorin Sav, Hyowon Lee, Alan F. Smeaton, and Noel E. O'Connor

Centre for Digital Video Processing,
Dublin City University, Glasnevin, Dublin 9, Ireland
sorinsav@eeng.dcu.ie, hlee@computing.dcu.ie,
asmeaton@computing.dcu.ie, oconnorn@eeng.dcu.ie

Abstract. This paper presents a system for video shot retrieval in which shots are retrieved based on matching video objects using a combination of colour, shape and texture. Rather than matching on individual objects, our system supports sets of query objects which in total reflect the user's object-based information need. Our work also adapts to a shifting user information need by initiating the partitioning of a user's search into two or more distinct search threads, which can be followed by the user in sequence. This is an automatic process which maps neatly to the ostensive model for information retrieval in that it allows a user to place a virtual checkpoint on their search, explore one thread or aspect of their information need and then return to that checkpoint to then explore an alternative thread. Our system is fully functional and operational and in this paper we illustrate several design decisions we have made in building it.

1 Introduction and Background

The continuous expansion of video archives has resulted in an increasing demand for effective information management systems. Browsing keyframes or fast-forward or automatic summarisation are all useful tools for navigating small amounts but large volume search capabilities are also required as archive sizes grow. Current approaches to searching use either (1) the ASR text or OCR text from still frames, or (2) match an external image or an existing keyframe against shot keyframes using low-level features like colour histograms or texture or edges or (3) automatically assign concept features such as indoor, outdoor, faces, dialogue, building, landscape, camera zooming, etc. and use these to filter shots for subsequent browsing. Each have advantages and each can successfully address some types of video retrieval [1]. However, often our information need when we search is for a specific object such as a search for a shot containing a motorbike or car, or a shot containing a horse. In such cases the spoken dialogue (ASR) may not describe what is on camera, overall content in a sample keyframe or image may be totally different to one containing our target object and thus overall colour, texture, edges etc. will be different, and there may not be an available concept feature detector for the object we are seeking. In such cases we need to search for actual objects, which is the focus of this paper.

Of course object search is not a panacea and works best when used as one of an available set of search tools including text, image match and feature filtering, all combined with a usable video browsing interface, but in this paper we concentrate on object retrieval as the retrieval tool. We have built a retrieval tool for video shot retrieval which retrieves based on objects and not based on the other modalities. The purpose here is not to do simple matching of an object from a query image against objects from a video keyframe but to use the selection of a *set of objects* in a query as the basis for retrieval. The purpose of our research is to explore object-based shot retrieval more than just object matching and as we will see this demands the formation and use of sets of query objects.

In the task of automatically segmenting and indexing objects in image/video content, the main difficulty is the diverse manifestation of an object in the image/video regardless of the object's inherent visual features such as colour, shape and texture. Due to factors such as different lighting conditions, different angles taken by the camera, and the degree and types of occlusions that often occur on objects, this makes the actual segmentation of an object as well as labelling the segmented object, for example a car, extremely difficult. This same problem of diverse manifestations of an object also occurs when a searcher has to give examples of an object during query formulation.

With this problem as the central issue, one workaround solution we have been exploring is to use ostensive relevance feedback, which takes a human user's judgements on object definitions, in retrieving objects. There is a long history of experimentation and successful use of relevance feedback in text-based information retrieval. This has included short-term modelling of a user's information need by dynamically updating the user's query formulation in mod-search as well as long-term modelling of user's needs by profiling his/her interests over time.

In this paper, we present an interactive, object-based search system that uses a novel, adaptive query formulation mechanism. As query formulation is the key element for getting feedback from the user in our approach, the system we have built incorporates a user interaction strategy in which a user can interact with segmented objects by way of highlighting them, selecting them, and then using them in subsequent query formulation.

The novelty of our work lies in using automatic query branching into an ostensive relevance feedback framework as a means to provide the user with knowledge about the distribution of object features in the video collection. This is a two way feedback where the system is instructed about the relevance of retrieved objects and the user receives explicit indications about the mapping of the query into the feature space. By being aware of the ramifications which a query has on the collection space, the user can better adapt the query and their feedback to more accurately select query objects relative to their information need.

The paper is organised as follows: in Section 2 we give an overview of ostensive relevance feedback as used for video object shot retrieval, Section 3 describe the algorithm used for video object retrieval and Section 4 presents the feature descriptors for video objects used in our system. The design of the object-user interaction mechanism and the front-end user interface is described in Section 5.

Section 6 concludes the paper with our plans for extending the system's capability and further refining the user-interface.

2 Ostensive Relevance Feedback Applied to Video Shot Retrieval

The process of information retrieval is an inherently uncertain one [2]. Users may find difficulty expressing their information need into an appropriate request for the retrieval system and they may not have a good idea of what information is available for retrieval. The concept of relevance feedback had arisen from the observation that although searchers have difficulties formulating retrieval queries, they can recognise relevant documents when the documents presented contain useful information. Relevance information can be exploited quantitatively - retrieving more documents similar to the relevant ones - and qualitatively - ranking higher documents that better match relevant ones [2].

The ostensive model of cognition described in [3] relates changes in the knowledge state of a user in response to information encountered during information seeking activities. The core components of the model are shown in Figure 1.

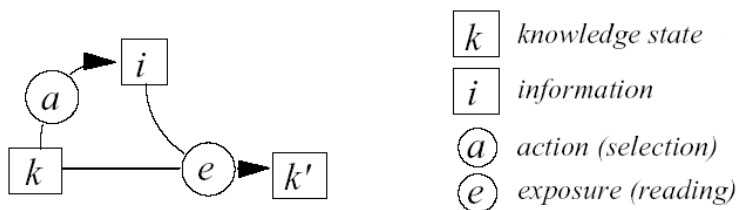


Fig. 1. The updating of a knowledge state through the selection of, and subsequent exposure to information, taken from [3]

According to this model the *knowledge state* k updates to *knowledge state* k' subject to *selection action* a and *exposure* e . The update process can continue into an iterative loop with state k updating to state k' then to k'' and so on. Mapping this model to a retrieval system we conclude that the *knowledge states* $k, k' ..$ are specific to both users and system (for every *knowledge state* k of the user there is a virtual *knowledge state* k of the system), the *action* a is performed by the retrieval system when retrieving the document i and the *exposure* e is performed by the user. Going further into the approach taken in [3] that discusses only the user perspective, we consider simultaneously the system perspective. We argue that *action* a - the retrieval of a document by the system - and *exposure* e - the feedback returned by the user - are observable, but the *knowledge states* k are non-observable. The user can see that a particular document has been retrieved but may not understand the system state that triggered the selection of that specific document. Similarly, the system, receiving feedback, is instructed

about the relevance of the document without being informed in which way that document is relevant to the user.

In terms of video shot retrieval based on objects, there are many parallels with this model in that users move from one knowledge state to another based on exposure to some video clip or shot. We would argue that it is even more the case in video shot retrieval, based on objects, that users will not understand the system’s reasoning as to why one shot may have been retrieved and that the system should be given something more than just yes/no relevance judgments on shots/objects and that relevance feedback should be faceted where possible. In the work we report in this paper we shall show how we achieve just that.

Researchers in the field of text retrieval have experimented with explanations as a technique “to reduce the conceptual gulf between how the system operates and how the user thinks the system operates” [4], [5]. In retrieval systems the explanatory power has been traditionally exploited by two models: the dialogue model [5], and domain knowledge representation [6]. The dialogue model controls what is to be explained and at what stage, whereas the domain knowledge determines the content of the explanation. However both models have drawbacks: domain knowledge representation is hard to achieve on heterogenous data collections [4], and there is strong indication that most users do not follow the search strategies proposed to them by the dialogue model [7].

Our system makes use of implicit explanations by visually showing the query documents (video objects) grouped in clusters based on their feature similarity. This visual representation provides the user with a intuitive explanation regarding the distribution of the relevant documents in the searched collection. To build a query the user can indicate positive and negative examples of video objects. By grouping the query objects into clusters, the system is suggesting to the user that their information need has actually diversified into two or more distinct categories of object retrieval which has already been differentiated by the system. This reflects the case of a user wishing to explore two aspects or branches of their query, which our system can support as we show later, and this neatly maps onto the ostensive model of retrieval where a user is encouraged to explore one aspect freely until it is exhausted and then return to this point and launch an exploration into the second aspect.

3 Video Shot Retrieval Algorithm

In this section we give an outline of our algorithm for video shot retrieval based on multiple example query objects. Once the user had provided (through relevance feedback) a set of objects as an indication of the objects they wish to retrieve, these are analysed in terms of colour, shape and texture. Considering these features as independent of each other we define an object-to-object similarity measure S_{object} as:

$$S_{object}(i, j) = \alpha S_{colour}(i, j) + \beta S_{shape}(i, j) + \gamma S_{texture}(i, j) \quad (1)$$

where α , β and γ are normalisation factors for the colour S_{colour} , shape S_{shape} and texture $S_{texture}$ similarity measures. For each feature the corresponding

similarity measure is independently computed and adjusted to better match the positive examples provided by the user through relevance feedback. The α , β and γ factors are directly proportional to the number of positive examples provided by the user for each of the respective features.

For each feature we assume that the positive examples can be modelled by a Gaussian mixture model where mixture component is a Gaussian with mean μ and covariance matrix Σ :

$$p(\varepsilon|j) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\varepsilon-\mu_j)^T \Sigma_j^{-1}(\varepsilon-\mu_j)} \quad (2)$$

The mean μ and variance Σ are estimated from examples labelled as relevant (positive). We consider the optimal model to be the Gaussian mixture with the minimum number of components that correctly classifies the labelled set of objects. This means that any object labelled as a positive instances is within one standard deviation away from the mean of at least one Gaussian component and that no object labelled as a negative instances is within two standard deviations away from the mean of any Gaussian component. The expectation-maximisation (EM) algorithm [8] is employed to estimate the density probability functions of the Gaussian mixture.

At this point there is a vector of parameters (μ, Σ) for the Gaussian mixture that models each feature (colour, shape, texture).

$$\begin{aligned} \bar{v}_{colour} &= ((\mu_{colour}^{(1)}, \Sigma_{colour}^{(1)}), \dots, (\mu_{colour}^{(n)}, \Sigma_{colour}^{(n)})) \\ \bar{v}_{shape} &= ((\mu_{shape}^{(1)}, \Sigma_{shape}^{(1)}), \dots, (\mu_{shape}^{(m)}, \Sigma_{shape}^{(m)})) \\ \bar{v}_{texture} &= ((\mu_{texture}^{(1)}, \Sigma_{texture}^{(1)}), \dots, (\mu_{texture}^{(p)}, \Sigma_{texture}^{(p)})) \end{aligned} \quad (3)$$

The components of these vectors are then combined such that each component of the colour vector is grouped with each component of the shape and texture vectors, constructing a query triplet.

$$query_{(i,j,k)} = ((\mu_{colour}^{(i)}, \Sigma_{colour}^{(i)})(\mu_{shape}^{(j)}, \Sigma_{shape}^{(j)})(\mu_{texture}^{(k)}, \Sigma_{texture}^{(k)})) \quad (4)$$

Each query triplet is a possible search direction and is displayed in the user interface by grouping together the video objects that belong to this triplet (see Section 5). There is a possibility of the number of queries growing exponentially with the number of features and so we limit the expansion of triplets by introducing a ‘‘mixture expansion factor’’ that constrains the increase in the number of components in the Gaussian mixture. The expansion factor is $1/N$, where N is the number of existing component in the Gaussian mixture. As N increases, the expansion factor becomes smaller therefore inhibiting the addition of new components. The user has the option to select one of the displayed queries (group of objects) as the active query in the next iteration. In one sense what we have done is to automatically categorise user query objects where each query category could represent a set of objects which are similar to each other but dissimilar to

other query objects. So, for example, if a user is searching for motor car objects then one category could be red VW Beetle objects and another category could be white jeep objects and the two categories of objects will have different colours (red or white) and shapes (Beetles are more curved in shape than jeeps), though textures may be similar.

In the next retrieval step we calculate the similarity distance from the mean μ and variance Σ of each feature in the active query to the features of the objects in the collection. The estimation-maximisation and query construction steps are repeated when new examples are labelled by the user.

4 Feature Descriptors for Objects

The features selected for image representation are colour, shape and texture as they are directly related to human perception and independent of each other. In our system the features describe only the image foreground (segmented object). We realise that image background conveys important information as well, but we do not consider this in this investigation.

4.1 Colour Representation

To represent colour we adopted the MPEG-7 Dominant Colour Descriptor (DCD) [9], which is used by many retrieval systems. The recommended distance to be used with DCD is [10] :

$$D_{DCD}(Q, I) = \left(\sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2\alpha_{1i,2j} p_{1i} p_{2j} \right)^{1/2} \quad (5)$$

where N is a set of colour vectors c_i , and p_i their percentages. The similarity coefficient $\alpha_{k,l}$ between two RGB color vectors c_k and c_l is calculated as:

$$\alpha_{k,l} = \begin{cases} 1 - \frac{D_{k,l}}{D_{max}}, & D_{k,l} \leq T_d \\ 0, & D_{k,l} > T_d \end{cases} \quad (6)$$

In expression 6 $D_{k,l} = \| c_k - c_l \|$ represents the Euclidian distance between two colour vectors. $T_d = 20$, $\alpha = 1$, and $D_{max} = \alpha T_d = 20$, follow the values given in [11].

4.2 Shape Representation

Shape description and similarity is an extremely complex research topic. The 2D projection on the image plane, elastic deformations of the object, and diversity of shapes in which instances of the same semantic object appear in the real world are common problems that must be considered for shape similarity. In our work, we use a relatively simple shape descriptor corresponding to the compactness

moment γ [12], defined by Equation 7. This is a simple and robust descriptor that can indicate a degree of shape similarity.

$$\gamma = \frac{P_2}{4\pi A} \quad (7)$$

where A is the area and P perimeter of the video object defined as:

$$P = \sum_{i=1}^{N-1} \|x_{i+1} - x_i\| + \|x_N - x_1\| \quad (8)$$

4.3 Texture Representation

In our system texture is represented with the MPEG-7 Texture Browsing Descriptor [9]. This descriptor is expressed as a set of 24 Gabor wavelets [13] $g_{m,n}(x, y)$ (6 orientations, 4 scales) obtained by appropriate rotations and dilations of the a two dimensional Gabor function:

$$\begin{aligned} g_{m,n}(x, y) &= a^{-m}G(x', y'), \quad a > 1 \\ x' &= a^{-m}(x \cos \theta + y \sin \theta) \\ y' &= a^{-m}(-x \sin \theta + y \cos \theta) \end{aligned} \quad (9)$$

where $\theta = n\pi/K$, K is the total number of orientations and a^{-m} is the scale factor. $G(x', y')$ is the Fourier transform of a two dimensional Gabor $g(x, y)$ function:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi jW \right] \quad (10)$$

Given an image $I(x, y)$ its Gabor wavelet transform is then defined as:

$$W_{m,n}(x, y) = \int \int I(x', y') g_{m,n}^* (x - x', y - y') dx dy \quad (11)$$

where $*$ indicates the complex conjugate and $g_{m,n}$ are the Gabor wavelets. It is assumed that the local texture regions are spatially homogeneous, and the mean $\mu_{m,n}$ and the standard deviation $\sigma_{m,n}$ of the magnitude of the transform coefficients are used to represent the region classification for retrieval purposes [13]:

$$\begin{aligned} \mu_{m,n} &= \int \int |W_{m,n}(x, y)| dx dy \\ \sigma_{m,n} &= \sqrt{\int \int (|W_{m,n}(x, y)| - \mu_{m,n})^2 dx dy} \end{aligned} \quad (12)$$

The resulting vector has $\mu_{m,n}$, $\sigma_{m,n}$ feature components. Then the distance between two patterns i and j in the texture space [13] is defined as:

$$\begin{aligned} d(i, j) &= \sum_m \sum_n d_{m,n}(i, j) \\ d_{m,n}(i, j) &= \left| \frac{\mu_{m,n}^{(i)} - \mu_{m,n}^{(j)}}{\alpha(\mu_{m,n})} \right| + \left| \frac{\sigma_{m,n}^{(i)} - \sigma_{m,n}^{(j)}}{\alpha(\sigma_{m,n})} \right| \end{aligned} \quad (13)$$

where $\alpha(\mu_{m,n})$ and $\sigma(\mu_{m,n})$ are the standard deviations of the respective features over the entire collection and are used to normalise the individual feature components.

5 User Interface and System Interaction

In this section we focus on the front end side of the system we have developed to allow the user to select query objects and to include query branching where the system offers two or more diverging queries for the user to pursue (red VWs and white jeeps in the example earlier).

We start with a description of the design scheme we developed to allow the user to browse and specify a particular object within an image content and use only that object for subsequent querying; then we describe how this scheme has been incorporated into an overall interface in which the interactive search stages (browsing, collecting relevant objects, querying based on the objects and re-querying) are implemented.

5.1 Interacting with Objects

Objects automatically detected by the system should be visible to the user in some way, so that s/he could see what possible further interaction can be done with it.

Figure 2(a) shows a keyframe, in this case showing a white car. An oval button on the right represents the detection of an object within the image with its three low-level features. If there is more than one object detected in the image, there will be a button for each.

In Figure 2(b), the user highlights the object of interest by selecting the button though in our current implementation there is a maximum of one object per

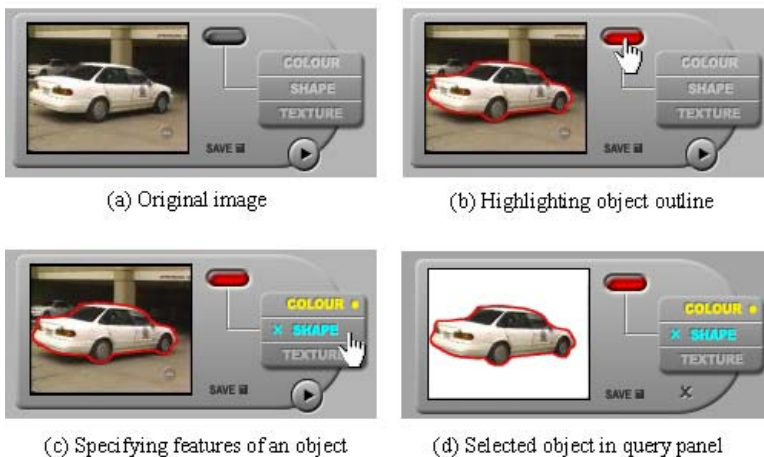


Fig. 2. Object visualisation and interaction

keyframe. In Figure 2(c), after selecting an object the user then can specify which low-level features (colour, shape or texture) of the specified object s/he is interested in. Each of the feature buttons toggles between positive, negative or neutral for each feature of the object. Once feature indications are specified, the user can copy this object (and its specified features) to the query panel as shown in Figure 2(d) where the image contains only the specified object with the background stripped away. The feature specification for this object will be now used for relevance feedback.

5.2 Relevance Feedback Using Objects

The main features of the interface to our object-based video retrieval system are for the user to:

- Browse initial set of objects;
- Specify particular features of an object to use for relevance feedback;
- Browse a number of user-selected objects and their specified features to adjust, remove, and add to the set of query objects;
- Trigger retrieval based on the specified features of the query objects;
- Browse retrieved objects and use some as additional feedback;
- Save relevant objects in a separate folder

In addition to the above, an important feature of our system is to allow the user to view how his/her relevance feedback and set of query objects is semantically consistent/inconsistent by showing clusters within the set of query objects. If this set of query objects is not visually consistent, using all of this feedback for retrieval will confuse the system and lower the retrieval accuracy. This is similar to adding very visually different image examples in Query-By-Example systems. Although a syntactically legitimate action by the user, this behaviour results in degraded retrieval and thus contributes negatively to the interaction. Thus, if the system can split the relevance feedback and set of query objects into semantically coherent query object groups and present them to the user, s/he can identify this and “branch” the query into two or more and then focus on only one of the groups at a time. This maps back nicely to the ostensive model of retrieval where a user wishes to pursue two or more “lines of enquiry” but can only do one at a time. With our approach, where the system has partitioned the search into two or more distinct clusters, one can be pursued as the set of query objects while the other cluster(s) is put on hold and returned to at a later stage. A worked example illustrates this.

Figure 3 shows a screen shot from the system in which the interface is divided into 4 columns. The first column presents an initial set of images in the representation format described in Section 5.1. The user browses this set of images, views objects and specifies features, then adds some of the objects to the “QUERY OBJECTS” panel (2nd column). A similar interface facility can be found in numerous experimental image and video retrieval systems in which the user can select example images to be used for subsequent queries as a mechanism for relevance feedback, as in [15], [16], [17], [18], [19], however, unlike these systems,

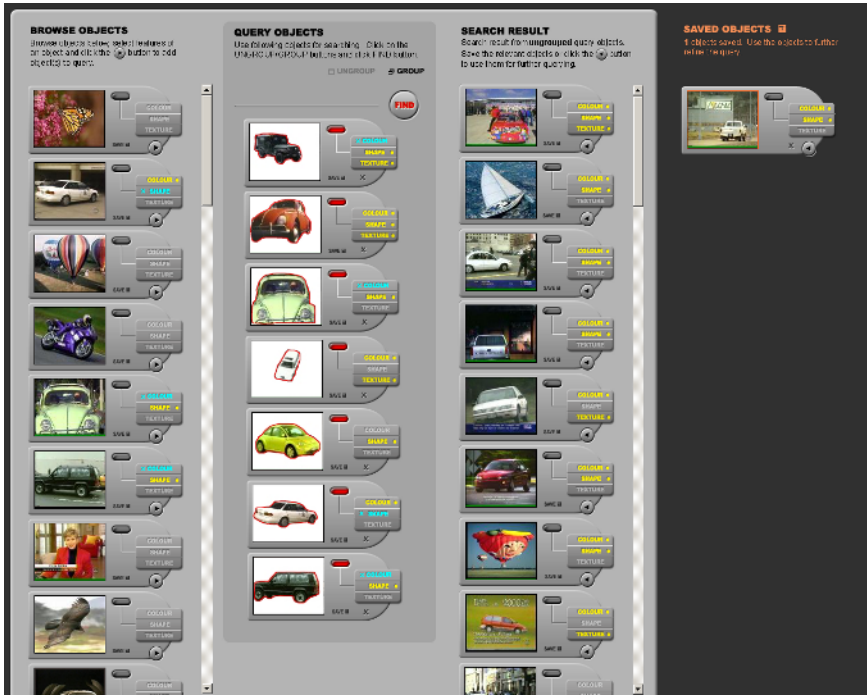


Fig. 3. Overall interface - query panel (2nd column) contains the 7 objects that the user selected during the interaction, and the 3rd column shows the search result based on the all 7 objects

the added examples in our system are objects, not a whole image or an image region. Figure 3 currently shows 7 objects added to the query panel. Clicking on the “FIND” button triggers retrieval based on the 7 objects and the positive, negative or neutral indicators of their features, and the result is presented on the “SEARCH RESULT” panel (3rd column). If a relevant object is found in the search result, the user can save it to the “SAVED OBJECTS” panel (4th column). The user can also add more objects to the query panel from the search result, or from the saved object panel. As the user browses, searches and saves more and more relevant objects, s/he can collect more relevant objects into the query panel.

At the top of query panel (2nd column), the user can click on the “GROUP” button to view how the system can internally split objects in the query panel. This split of query objects is displayed in Figure 4.

In Figure 4, the 7 objects the user added to the query have been split into 2 groups according to the system’s clustering algorithm. The user can now see how s/he has been adding objects of two different types: in the 1st group (top 4 objects in the 2nd column), the object characteristics indicate white colour, more square shaped vehicles such as a white jeep and in the 2nd group (bottom 3 objects in the 2nd column), the object characteristics indicate red, round shaped vehicles such as a VW Beetle, quite different from the one formulated in the first

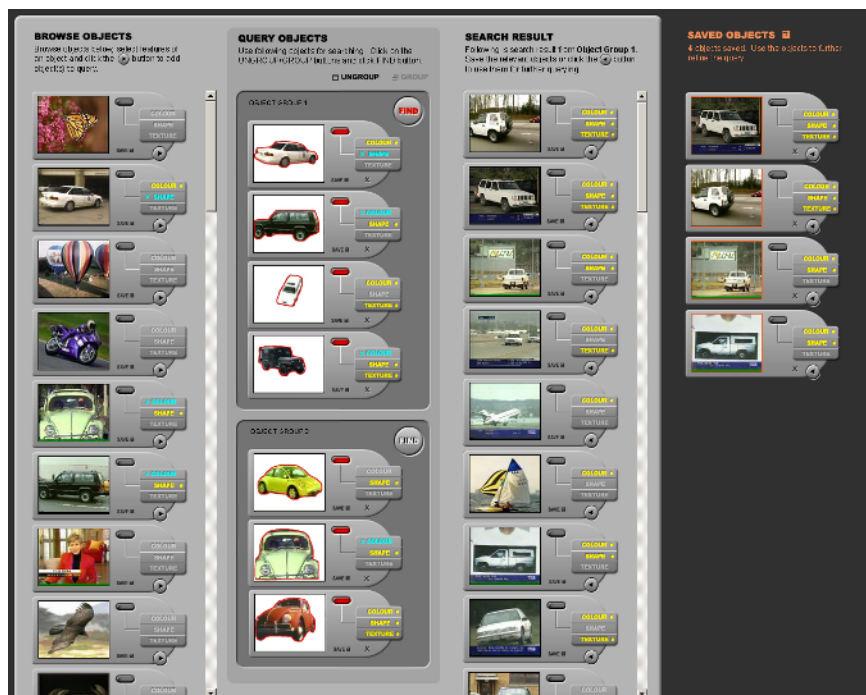


Fig. 4. Query objects are split into 2 groups as the result of the system’s internal clustering. The user can now pursue searching either or both of the query groups.

group. As this split among the added objects is now revealed to the user, s/he can decide to focus on searching for only one type of object (either 1st or 2nd group) to find more objects that are like only either of the groups. In Figure 4 the user searched on the 1st group by clicking on the “FIND” button of that group, and the top few search results show white cars with squared edges. The fact that the search result is from the 1st group of objects is indicated by the “FIND” button in that group and mentioned at the top of the search result. At this stage, when the user adds another object into the query panel it will be automatically inserted into the 1st group if the added object is deemed to be in the same cluster as the 1st group; or inserted into the 2nd group if deemed more similar to 2nd cluster; or as a separate, 3rd group in the case that it is far from the feature space of the either groups.

In this way, the user can see semantic clustering of query objects as s/he adds and specifies the features of objects, and can conduct a more multi-threaded search by pursuing one of the clusters of query objects at a time. Inconsistent relevance feedback is still a legitimate action by the user but our system is adaptive in that it suggests a better way of searching by automatically splitting the relevance feedback history into semantically coherent clusters so that the user can continue with a more consistent subset of his/her own saved feedback objects and can search query object clusters, one at a time. As mentioned earlier, this

maps neatly to one aspect of the ostensive model for retrieval where a user is confronted with two distinct threads to their search which they wish to pursue in sequence, both falling under the one information need. By automating the detection of these threading or branch-off points and maintaining both such threads as separate, live searches, the user is encouraged to follow his/her own instincts if these match the threads suggested by the system.

6 Conclusions

In this paper we introduced an object-based video search system that features interactive query formulation using the colour, shape and texture of an object, and through iteration of query/browsing, the system incrementally improves modelling of video objects. The actual segmentation of objects from keyframes in our system was semi-automatic and supervised in order to provide accurate object sets and to better illustrate our retrieval approaches in which the matching among objects (i.e. relating all similar objects in the database) can be helped using the user's query formulation history as feedback.

The status of our work is that we have build the retrieval system described in this paper and we have a collection of video with 650 semi-automatically segmented objects, we have completed the user interface as described here, we have completed some initial user testing and we are starting a more comprehensive interactive user testing and evaluation.

We are also working on several improvements including making object segmentation from each keyframe fully-automatic. Segmenting more than one object from each keyframe is also part of our future work; our user interface accommodates interaction with more than one object in a single keyframe (by way of multiple buttons). Currently a keyframe from a shot is used to segment objects however a more complete solution would be to use all frames within the shot, which could further provide additional information on the object from its movement and trajectory rather than from just the keyframe.

Acknowledgments

The support of the Enterprise Ireland Informatics Initiative is gratefully acknowledged. Part of this work was supported by Science Foundation Ireland under grant 03/IN.3/I361.

References

1. Smeaton, A.F., Over, P. and Kraaij, W. : TRECVID: Evaluating the Effectiveness of Information Retrieval Tasks on Digital Video. in: *Proceedings of the 12th ACM International Conference on Multimedia 2004*, pp. 652-655, New York, NY, 15-16 October 2004.
2. Ruthven, I. and Lalmas, M. : A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowledge Engineering Review*, Vol. 18, pp. 95-145, 2003.

3. Campbell, I. and van Rijsbergen, C.J. : The Ostensive Model of Developing Information Needs. In: *Proceedings of the 2nd International Conference on Conceptions of Library and Information Science*, CoLIS 2, Copenhagen, Denmark, 1996.
4. Ruthven, I. : On the Use of Explanations as a Mediating Device for Relevance Feedback. in: *Proceedings of the 6th European Conference on Digital Libraries, ECDL 2002*, Lecture Notes in Computer Science, Rome, 2002.
5. Belkin, N.J. : On the Nature and Function of Explanation in Intelligent Information Retrieval. in: *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Grenoble, 1988.
6. Cawsey, A.: Explanation and Interaction: the Computer Generation of Explanatory Dialogues. MIT Press (The ACL-MIT Press Series in Natural Language Processing), 1992.
7. Dennis, S., McArthur, R. and Bruza, P. : Searching the WWW Made Easy ? The Cognitive Load imposed by Query Refinement Mechanisms. in: *Proceedings of the Third Australian Document Computing Symposium*, 1998.
8. Moon, T. K. : The Expectation-Maximisation Algorithm. *IEEE Signal Processing Magazine*, pp. 47-60, November 1996.
9. Salambier, P. and Smith, J.R. : MPEG-7 Multimedia Descriptions Schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, pp. 748-759, June 2001.
10. Manjunath, B. , Salambier, P. and Sikora, T. : Introduction to MPEG-7: Multimedia Content Description Interface. Wiley, New York, USA, 2002.
11. Kushki, A., Androustos, P., Plataniotis, K.N. and Venetsanopoulos, A.N. : Query Feedback for Interactive Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, Issue 5, pp. 644-655 , May 2004.
12. Theodoridis, S. and Koutroubas, K. : Pattern Recognition. Academic Press, 1999.
13. Manjunath, B.S. and Ma, W.Y. : Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, Issues 8, pp. 837-842, August 1996.
14. Ruthven, I., Lalmas, M. and van Rijsbergen, C.J. : Ranking Expansion Terms Using Partial and Ostensive Evidence. in: *Proceedings of the 4th International Conference on Conceptions of Library and Information Science*, CoLIS 4, Seattle, 2002.
15. Lu, Y., Hu, Ch., Zhu, X., Zhang, H.J. and Yang, Q. : A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems, in: *Proceedings of ACM Multimedia*, pp. 31-37, 2000.
16. Heesch, D. and Rüger, S. : Three Interfaces for Content-Based Access to Image Collections. in: *Proceedings of the International Conference on Image and Video Retrieval*, CIVR 2004, Dublin, pp. 491-499, 2004.
17. Worring, M., Nguyen, G.P., Hollink, L., van Germert, J. and Koelma, D.C. : Interactive Search Using Indexing, Filtering, Browsing, and Ranking. in: *Proceedings of the TRECVID Workshop*, Gaithersburg, Maryland, 15-16 November, 2004.
18. Cooke, E., Ferguson, P., Gaughan, G., Gurrin, C., Jones, G., Le Borgne, H., Lee H., Marlow, S., McDonald, K., McHugh, M., Murphy, N., O'Connor, N., O'Hare, N., Rothwell, S., Smeaton, A.F. and Wilkins, P. : TRECVID 2004 Experiments in Dublin City University. in: *Proceedings of the TRECVID Workshop*, Gaithersburg, Maryland, 15-16 November, 2004.
19. Carson, C. and Greenspan, H. : Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, Issue 8, 2002.

Learning User Queries in Multimodal Dissimilarity Spaces

Eric Bruno, Nicolas Moenne-Loccoz, and Stéphane Marchand-Maillet*

Viper group,
Computer Vision and Multimedia Laboratory, University of Geneva
`eric.bruno@unige.ch`

Résumé. Different strategies to learn user semantic queries from dissimilarity representations of audio-visual content are presented. When dealing with large corpora of videos documents, using a feature representation requires the on-line computation of distances between all documents and a query. Hence, a dissimilarity representation may be preferred because its offline computation speeds up the retrieval process. We show how distances related to visual and audio video features can directly be used to learn complex concepts from a set of positive and negative examples provided by the user. Based on the idea of dissimilarity spaces, we derive three algorithms to fuse modalities and therefore to enhance the precision of retrieval results. The evaluation of our technique is performed on artificial data and on the annotated TRECVID corpus.

1 Introduction

Determining semantic concepts by allowing users to iteratively refine their queries is a key issue in multimedia content-based retrieval. The relevance feedback loop allows to build complex queries made out of positive and negative documents as examples. From this training set, a learning process should then extract relevant documents from feature spaces. Many relevance feedback techniques have been developed that operate directly in the feature space [4, 15, 17, 20].

Describing content of videos requires to deal in parallel with many high-dimensional feature spaces expressing the multimodal characteristics of the audiovisual stream. This mass of data makes retrieval operations computationally expensive when dealing directly with features. The simplest task of computing the distance between a query and all other elements becomes infeasible when involving tens of thousands of documents and thousands of feature space components. This problem is even more sensible when the similarity measures are complex functions or procedures, such as prediction functions for temporal distances [3] or graph exploration for semantic similarities [14]. A solution to allow on-line interaction would be to compute off-line dissimilarity relationships between elements and to use the dissimilarity matrices or distance-based indexing structures [5] as an index for retrieval operations.

* This work is funded by the Swiss NCCR (IM)2 (Interactive Multimodal Information Management).

Another aspect to prefer similarities rather than features is the multimodal fusion problem. Dealing directly with multimodal and heterogeneous features imposes to build complex learning setup that need to take into account every specific properties of features [15, 18]. On the opposite, similarity measures provide us with an homogeneous framework where the learning process consists in combining various distance measurements, whatever the nature and the variety of the features involved into the description.

The problem is then to find distance-based solutions that go beyond the classical k -NN approaches [2, 10] in order to perform discriminative classification that provides effective retrieval of semantic concepts. Pekalska *et al* [13] have proposed dissimilarity spaces where objects are represented not by their features but by their relative dissimilarities to a set of selected objects. These representations seem to form a convenient approach to tackle the similarity-based indexing and retrieval problem.

In this paper, we show how dissimilarities can be used to build low-dimensional multimodal representation spaces where learning machines, (*eg* SVM), could operate and investigate several strategies to fuse the modalities. Our thorough evaluation on both artificial data and the TRECVID 2003 corpus shows that multimodal dissimilarity spaces allow to perform effective retrieval of video documents using real time interaction.

2 Query-Dependent Dissimilarity Space

In the proposed framework, users formulate complex queries by iteratively providing positive and negative examples in a relevance feedback loop. From this training data, the aim is to perform a real-time dissimilarity-based classification that will return relevant documents to user. We present in the following the dissimilarity space introduced by Pekalska *et al* in [13] and show how it can be adapted to provide us with a low-dimensional approximation of the original feature space where an efficient classification could be performed.

Let $d(\mathbf{x}_i, \mathbf{x}_j)$ be the distance between elements i and j according to their descriptors $\mathbf{x} \in \mathcal{F}$. \mathcal{F} expresses the (unavailable) original feature space. The dissimilarity space is defined as the mapping $\mathbf{d}(\mathbf{z}, \Omega) : \mathcal{F} \rightarrow \mathbb{R}^N$ given by :

$$\mathbf{d}(\mathbf{z}, \Omega) = [d(\mathbf{z}, \mathbf{x}_1), d(\mathbf{z}, \mathbf{x}_2), \dots, d(\mathbf{z}, \mathbf{x}_N)]. \quad (1)$$

The representation set $\Omega = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is a subset of N objects defining the new space. The new “features” of an input element are now its dissimilarity values with the representation objects. As a consequence, learning or classification tools for feature representations are also available to deal with the dissimilarities.

The dimensionality of the dissimilarity space is directly linked to the size of Ω , which controls the approximation made on the original feature space (such an approximation could be computed using projection algorithms like classical scaling [6]). Increasing the number of elements in Ω increases the representation accuracy. On the other hand, a well-chosen space of low dimension would be more effective for further learning processes as it avoids the *curse of dimensionality* problem and reduces computation load.

The selection of a « good » representation set may be driven by considerations on the particular learning problem we are dealing with. Let us denote the query as the set T of positive and negative training examples (respectively denoted \mathcal{P} and \mathcal{N} with $T = \mathcal{P} \cup \mathcal{N}$). As mentioned by Zhou *et al* [20], we are generally dealing with a $1 + x$ class setup with 1 class associated to positives and x to negatives. Dedicated algorithms, such as *Bias-Map* [20, 19], have been developed to tackle this classification problem. In our case, the choice of the set Ω offers us the possibility to turn the problem into a more classical formulation : Selecting the representation set as the set of positive examples \mathcal{P} turns the problem into a binary classification. Indeed, assuming that the positive examples are close to each other while being far from negative examples, the vectors $\mathbf{d}(\mathbf{z}_{i \in \mathcal{P}}, \mathcal{P})$ (*within* scatter) have norms lower than vectors $\mathbf{d}(\mathbf{z}_{i \in \mathcal{N}}, \mathcal{P})$ (*between* scatter), leading to a binarization of the classification, as illustrated in figure 1 with artificial data. In this particular case, the learning task does not consist anymore in estimating the circular distribution of the negative class but a simpler function that separate the positive class, close to the origin, to the rest of the space.

The second advantage of selecting \mathcal{P} as the representation set is that it readily induces to work in a low dimensional space of $p = |\mathcal{P}|$ components, where on-line learning processes are dramatically speed-up.

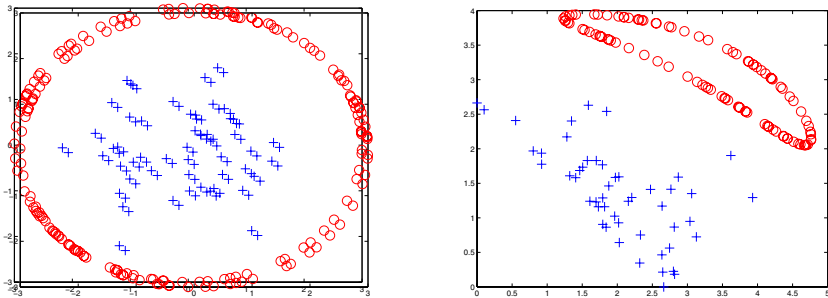


Fig. 1. The $1 + x$ class problem in feature space (left) and 2D dissimilarity space (right) where the representation objects are two points from the central class (cross)

3 Multimodal Dissimilarity Space

A multimodal description of multimedia data provides a number of feature spaces (one or more per modality). Each of them leads to a dissimilarity matrix containing pairwise distances between all documents, which are now referred by several dissimilarity measures that could be partially dependent. The success for interpreting a user query relies on the effective use of all of these information sources as well as their inter-dependencies. In the following, we discuss the different strategies to design a multimodal representation of data based on the dissimilarity spaces previously introduced.

We note d^{f_i} the distance measure applied to the feature space \mathcal{F}_i and assume that dissimilarity matrices are known for M feature spaces. Then, given

a set of positive examples \mathcal{P} , M monomodal spaces \mathbf{d}^{f_i} are built following the definition in 1.

A first way to fuse modalities is to consider that a possible multimodal dissimilarity would be the sum of all (normalized) monomodal distances. With this definition, the multimodal space \mathbf{d} is simply

$$\mathbf{d} = \mathbf{d}^{f_1} + \mathbf{d}^{f_2} + \dots + \mathbf{d}^{f_M} \in \mathbb{R}^p. \quad (2)$$

The dissimilarity space dimension is independent from the number of original feature spaces as it is always equal to p . This is a great advantage when M is large, but, on the other hand, one can object that the linear sum does not make sense to fuse features, especially when it deals with many sources of information. Moreover, the sum is sensitive to noisy and uninformative modalities, which might corrupt any classification operations.

To overcome these difficulties, we can consider that the fusion is carried out *a posteriori* by a classifier operating directly on multimodal components. The multimodal space is then the concatenation of all monomodal spaces, each element of which being represented by a multimodal dissimilarity vector

$$\mathbf{d} = \left[\mathbf{d}^{f_1 T}, \mathbf{d}^{f_2 T}, \dots, \mathbf{d}^{f_M T} \right]^T \in \mathbb{R}^{pM}. \quad (3)$$

This solution has the benefit to leave the fusion decision to the training process. However, it imposes to work in a higher-dimensional space where the estimation of the class distributions from a small training set may be less reliable.

Finally, we can adopt a more general fusion scheme where the input of the multimodal space is made of outputs of base classifiers. This solution is known as the *general combining classifier* [7, 16].

$$\mathbf{d} = \left[g_1(\mathbf{d}^{f_1}), g_2(\mathbf{d}^{f_2}), \dots, g_M(\mathbf{d}^{f_M}) \right]^T \in \mathbb{R}^M, \quad (4)$$

where $g_i(\cdot)$ denote the decision function of base classifier for the i th modality. The fusion algorithm is then split into two steps. First, individual classifiers are trained on their respective dissimilarity spaces. The classifier outputs are then used as input of a super classifier who takes the final fusion decision. This solution has the benefit to work in low-dimensional spaces but imposes $M + 1$ classifications, leading to a computational over-head.

The choice between one of these three strategies will be discussed in the light of the results exposed in section 5. The problem now is to define which classification algorithms with which parameters will be used to learn queries.

4 Classification

Many algorithms can be used to train a classifier that will learn semantic concepts. We have chosen to use an SVM because of its effectiveness and its flexibility in parametrization. The kernel selection and setting is a critical issue to successfully learn queries. It actually decides upon the classical trade-off between over-fitting

and generalization properties of the classifier and hence is very dependent of the considered representation space. Depending of the multimodal space used, we differentiate three setups for the classifier :

- sum of dissimilarity spaces (def. 2) : An RBF kernel $k(\mathbf{x}, \mathbf{y}) = e^{-(\mathbf{x}-\mathbf{y})^T \mathbf{A}(\mathbf{x}-\mathbf{y})}$ with $\mathbf{A} = \sigma^{-1} \mathbf{I} \mathbf{d}$ is used. The estimation of σ is based on a heuristic adapting the model to the query

$$\sigma = C \cdot \text{median}_i (\min_j \|\mathbf{d}_i^+ - \mathbf{d}_j^-\|^2) \quad (5)$$

The scale value is tuned to the median of all the minimum distances between the negative and the positive examples. In that way, the kernel becomes tighter as the two classes become closer to each other. The parameter C has been empirically set to 2.0.

- concatenation of spaces (def. 3) : The same RBF kernel is used but $\mathbf{A} = \text{diag}[\sigma_{f_1}, \dots, \sigma_{f_M}]$ so as to allow independent scaling for each modality. The scale vector $\sigma_{f_i} \in \mathbb{R}^p$ is constant with all values equal to the scale parameter σ_{f_i} computed for each monomodal space \mathbf{d}^{f_i} using the formula (5).
- hierarchical classification (def. 4) : Independent classification of monomodal spaces are done with an RBF kernel and a scale σ computed with (5). As super classifier, a sigmoid kernel is used with a scaling parameter C set to 0.1.

The three above definitions become clearly equivalent when only one modality is considered. In the following, for the clarity of the results, we denote them respectively as (2) SUM, (3) CONC and (4) HIER.

5 Experimentations

The following experimentations have been conducted to compare the three proposed multimodal fusion algorithms and to measure the efficiency of the approach in a real video retrieval application. For both artificial and real annotated data, the experimentation consists in making queries corresponding to concepts and measuring the Average Precision, AP as the sum of the precision at each relevant hit in the retrieved list divided by the minimum between the number of relevant documents in the collection and the length of the list. The retrieved list has 100 entries and the AP measure is averaged over 50 queries. The annotated positive examples are removed from the hit-list so that they are not taken into account when measuring performances (pessimistic evaluation). The Mean Average Precision (MAP) is the AP averaged over several concepts. We also used it when appropriate.

5.1 Video Database and Features

We have considered 60 hours of the annotated video corpus TRECVID-2003. This corpus is composed of CNN and ABC news. Videos are segmented into shots

and every shot has been annotated by several concepts. The speech transcripts extracted by Automatic Speech Recognition (ASR) at LIMSI laboratory [8] are also available.

We extracted the six following features from the 37'500 shots composing the corpus :

- Color histogram,
- Motion vector histogram,
- Local features [11],
- Word occurrence histogram (after stemming and stopping),
- Latent Semantic Indexing [1],
- Dominant audio features [9].

The distance measures used are Euclidean for color and motion histograms and intersection for word occurrence histograms. An approximation of the minimal matching distance is applied on local features to determine partial similarities [12], cosine distance is used for LSI and the audio similarity measure proposed in [9] is used for audio features.

5.2 Results

We first test the validity of the monomodal dissimilarity space defined in section 2. We compare the accuracy of the retrieval when the classification is performed in the color feature space of 64 dimensions and in the corresponding dissimilarity space. Let's note that the dissimilarity space's dimension is equal to the number of positive examples provided to the system. The figure 2 shows AP performances for two queries corresponding to two annotated concepts (*Basketball* and *Studio setting*). We can observe that learning in dissimilarity space yields much better performance than in feature space when the training set is small (typically less than 40 positive examples). After that, the performances become comparable into the two spaces. This very interesting behavior for small training set is due to the simplification of the classification problem implied by the dissimilarity space (see section 2).

The two following experiments consist in characterizing and comparing the fusion strategies on artificial data. The features are generated from statistical distributions and Euclidean distance is used to compute pairwise dissimilarities.

We first evaluate the discriminative power of the three strategies. The features are drawn from two multivariate centered Gaussian distributions so as to generate three dissimilarity spaces simulating three modalities. The positive class has a variance fixed to 1 whereas the variance of the negative class varies from 1 to 3.5. That way, the negative class is gradually surrounding the positive one. We then measure how fast the fusion algorithms are able to isolate the positive elements from the rest of the data. The figure 3 plots the Average Precision for our three fusion strategies SUM, CONC and HIER.

We observe that the CONC and HIER strategies show similar results and outperform the SUM scheme to discriminate between the two classes. It validates

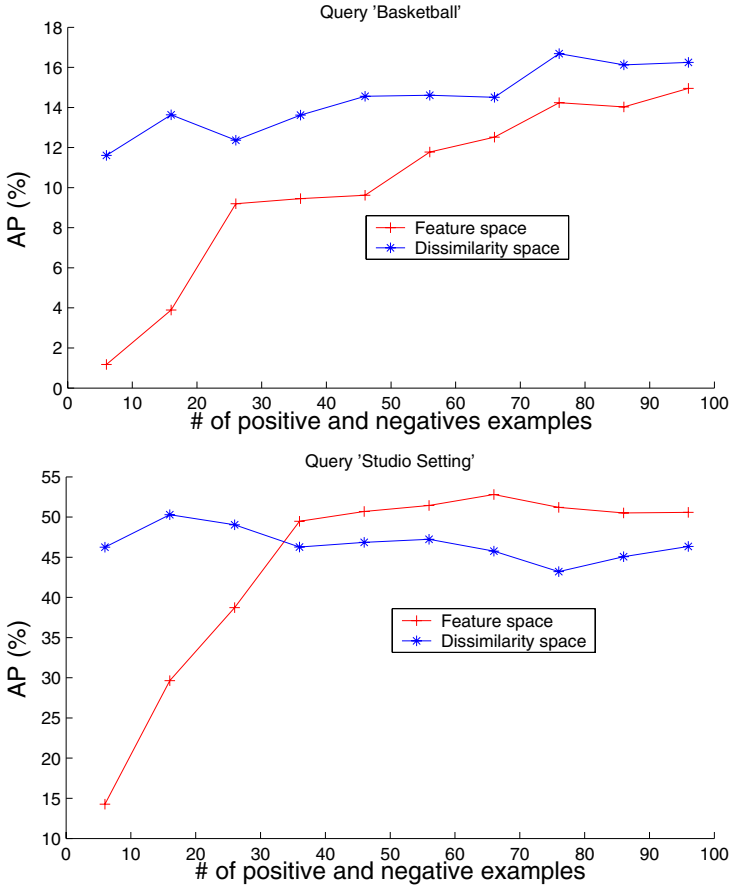


Fig. 2. Average precision when the classification is performed directly in the color feature space (cross) and in the corresponding dissimilarity space (square) as the number of positive and negative examples increases

our prior idea that the SUM fusion scheme is clearly not optimal to combine dissimilarities.

We evaluate now the strategies in term of robustness to uninformative modalities. The problem is simulated by 10 modalities where one is purely informative (the dissimilarities are set to 0 for elements belonging to the sought concept and 1 for the rest) and the nine others are just uniform noise. Figure 4 shows AP results when the amplitude of the noise rises from 0 to 10. Again we can observe that the SUM algorithm performs worse than the two others. But only the hierarchical classification is actually robust to noisy modalities.

The next step of the evaluation consists in studying on real data how the combination of modalities might improve the retrieval efficiency. The task is to retrieve shots that are annotated by a particular concept, using either monomodal and multimodal spaces and the three fusion algorithms. The figure 5 display

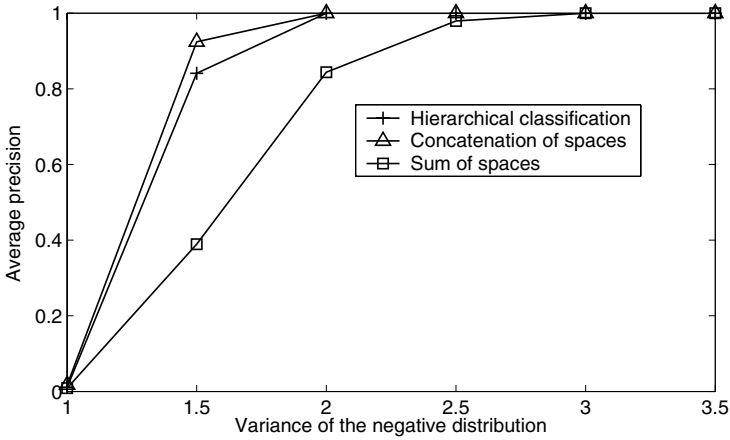


Fig. 3. Average precision vs. the variance of the Gaussian distribution corresponding to negative examples (the positive Gaussian distribution’s variance is set to 1). The query is composed of 10 positive and 10 negative examples randomly chosen.

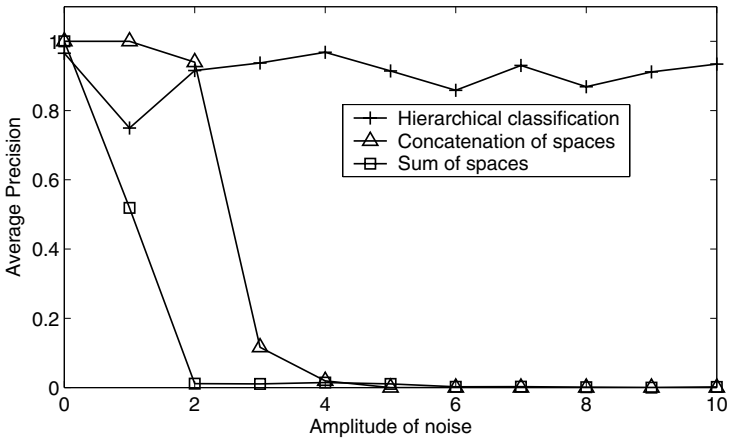


Fig. 4. Average precision on artificial data where one modality is purely informative while the 9 others are purely non-informative. The query is composed of 10 positive and 10 negative examples randomly chosen.

the MAP results (evaluated over ten semantic concepts) for every features and every fusion algorithms. These results are also compared to a random guess (e.g seeking hits at random within the database).

This result shows that color and ASR histograms are the best features to characterize concepts, but the addition of less relevant features improves dramatically the performances. Whatever the fusion strategy used, the AP is at least doubled when multimodal information is taken into account. Among the

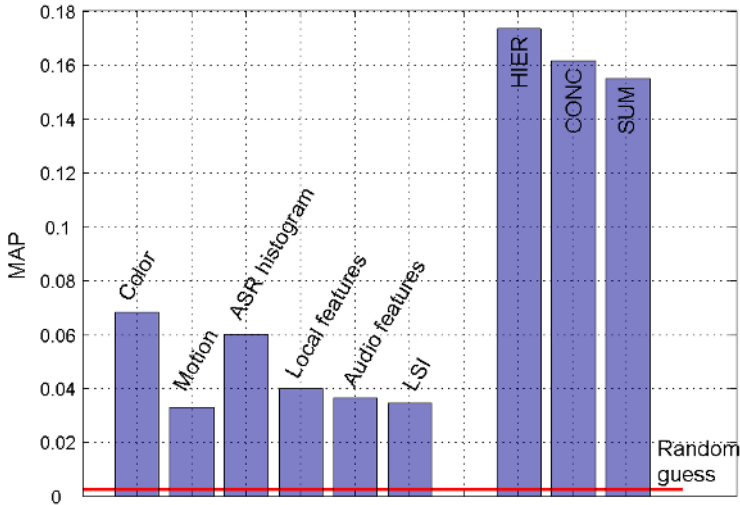


Fig. 5. MAP for monomodal and multimodal retrieval and for the three multimodal fusion strategies. The queries are composed of 10 positive and 10 negative examples.

fusion algorithms, HIER performs the best, followed by CONC and SUM as predicted by experiments on artificial data. However, the overall performances are of the same order (AP increases of 2% between SUM and HIER). It is indeed disappointing that the less discriminative and robust SUM strategy gives globally comparable results on real data. It is however worth noting that, from the point of view of optimization, SUM provides a well-posed and simple setup by minimizing both the dimensionality of the representation space and the number of classifications to operate. This simplicity is probably the reason of its good behavior.

In the last experiment, we simulate a user querying the system using relevance feedback. Initially, 2 positive and 5 negative examples are given to the system. A first hit-list is provided back where up to 5 new positive and negative examples (randomly selected) are added to the initial query. This query is resubmitted to the system providing a new hit-list. This is repeated until the twentieth iteration. As figure 6 shows, the MAP increases with the number of iterations until a maximum value is reached from where the addition of new examples does not improve anymore the classification accuracy. This saturation behavior illustrates how the users, by providing more and more examples can refine their queries until reaching the optimum of the classifier.

Finally, we give the computation load of each algorithm (Table 5.2) for various training sets and for 3 and 10 modalities. These times have been obtained on a PC PIV 2GHz (Matlab implementation). We see that the SUM strategy is effectively the fastest while the computation load of the hierarchical classification increases linearly with the number of modalities used.

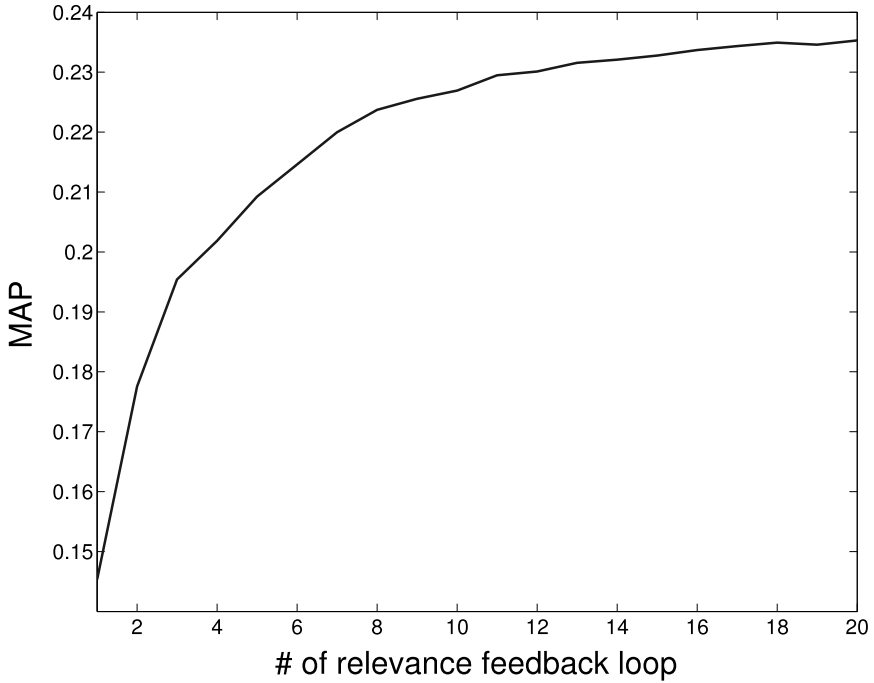


Fig. 6. Relevance feedback simulation. At each iteration, up to five positive and five negative examples are added to the query. HIER algorithm is used and MAP is evaluated over ten concepts.

Table 1. Computation load (in second)

	Training set		Algorithms		
	Positive ex.	Negative ex.	SUM	CONC	HIER
3 modalities	10	10	0.3	0.4	1.2
	20	10	0.5	0.8	1.9
	40	10	1.4	1.7	5.9
	10	40	1.2	1.3	4.2
10 modalities	10	10	0.3	0.6	3.4

6 Discussion

Our three dissimilarity-based multimodal fusion strategies are able to take benefit from low-level audio-visual descriptions of video documents and, as a consequence, to learn semantic queries from a limited number of input examples. Moreover, the fusion of the information sources performs better than considering modalities independently.

The design of the dissimilarity space as been achieved so as to simplify the classification problem while building a low-dimensional representation of the

data. As a result, queries on large databases are processed in near real-time which authorizes the use of feedback loop as a search paradigm.

In the light of the results, we are able to rank the efficiency of the three approaches for the retrieval task. Tests on artificial and real data have demonstrated the superiority of the hierarchical classification in terms of class discrimination and robustness to corrupted modalities. The computation over-head is however not negligible, especially when a large number of features is used. On the other hand, the experiments carried out on real data do not exhibit indisputable superiority between the three fusion strategies. It is actually interesting to note that the naive but fast SUM fusion is able to compete with more sophisticated methods and present a good trade-off between computation load and retrieval accuracy. However, to be completely valid, these results have to be confirmed when more modalities/features are involved in the fusion process.

Our future work will therefore consist in extracting new features that better characterize audiovisual content. It will enable us to properly evaluate the fusion algorithms and to determine the limits of the fusion schemes when a large number of features are used.

Références

1. M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4) :573–595, 1995.
2. Liudmila Boldareva and Djoerd Hiemstra. Interactive content-based retrieval using pre-computed object-object similarities. In *Conference on Image and Video Retrieval, CIVR'04*, pages 308–316, Dublin, Ireland, 2004.
3. Eric Bruno, Nicolas Moenne-Loccoz, and Stephane Marchand-Maillet. Unsupervised event discrimination based on nonlinear temporal modelling of activity. *Pattern Analysis and Application, special issue on Video Event Mining*, 2005. DOI : 10.1007/s10044-005-0242-9.
4. E. Y. Chang, B. Li, G. Wu, and K. Go. Statistical learning for effective visual information retrieval. In *Proceedings of the IEEE International Conference on Image Processing*, 2003.
5. E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquin. Searching in metric spaces. *ACM Computing Surveys*, 33(3) :273–321, September 2001.
6. T.F. Cox and M.A.A. Cox. *Multidimensional scaling*. Chapman & Hall, London, 1995.
7. R.P.W. Duin. The combining classifier : To train or not to train ? In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR'02*, volume II, pages 765–770, Quebec City, 2004. IEEE Computer Society Press.
8. J.L. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. *Speech Communication*, 37(1-2) :89–108, 2002.
9. J. Gu, L. Lu, H.J Zhang, and J. Yang. Dominant feature vectors based audio similarity measure. In *PCM*, number 2, pages 890–897.
10. D Heesch and S Rueger. Nnk networks for content-based image retrieval. In *26th European Conference on Information Retrieval*, Sunderland, UK, 2004.

11. Nicolas Moënne-Loccoz, Eric Bruno, and Stéphane Marchand Maillet. Interactive retrieval of video sequences from local feature dynamics. In *Proceedings of the 3rd International Workshop on Adaptive Multimedia Retrieval, AMR'05*, Glasgow, UK, July 2005.
12. Nicolas Moenne-Loccoz, Eric Bruno, and Stéphane Marchand-Maillet. Interactive partial matching of video sequences in large collections. In *IEEE International Conference on Image Processing (ICIP'05)*, Genova, Italy, 2005.
13. E. Pekalska, P. Paclík, and R.P.W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2 : 175–211, December 2001.
14. Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *14th International Joint Conference on Artificial Intelligence, IJCAI*, pages 448–453, Montreal, Canada, 1995.
15. J. R. Smith, A. Jaimes, C.-Y. Lin, M. Naphade, A. Natsev, and B. Tseng. Interactive search fusion methods for video database retrieval. In *IEEE International Conference on Image Processing (ICIP)*, 2003.
16. Y. Wu, E. Y. Chang, K.C-C Chang, and J.R Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of ACM Int. Conf. on Multimedia*, New York, 2004.
17. R. Yan, A. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *Proceedings of ACM Multimedia (MM2003)*, Berkeley, USA, 2003.
18. J. Yang and A.G. Hauptmann. Multi-modality analysis for person type classification in news video. In *Electronic Imaging'05 - Conference on Storage and Retrieval Methods and Applications for Multimedia*, San Jose, USA, Jan.
19. X.S. Zhou, A. Garg, and T.S. Huang. A discussion of nonlinear variants of biased discriminant for interactive image retrieval. In *Proc. of the 3rd Conference on Image and Video Retrieval, CIVR'04*, pages 353–364, 2004.
20. X.S. Zhou and T.S. Huang. Small sample learning during multimedia retrieval using biasmap. In *Proceedings of the IEEE Conference on Pattern Recognition and Computer Vision, CVPR'01*, volume I, pages 11–17, Hawaii, 2004.

Surface Features in Video Retrieval

Thijs Westerveld, Arjen P. de Vries, and Georgina Ramírez

CWI, INS1, PO Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract. This paper assesses the usefulness of surface features in a multimedia retrieval setting. Surface features describe the metadata or structure of a document rather than the content. We note that the distribution of these features varies across topics. The paper shows how these distributions can be obtained through relevance feedback and how this allows for adaptation of (content-based) search results for topic or user preference. An analysis of the distribution of surface features in the TRECVID collection indicates that they are potentially useful, and a preliminary feedback experiment confirms that exploiting surface features can improve retrieval effectiveness.

1 Introduction

Multimedia retrieval typically relies on either low-level content features, like colour or texture descriptors, or on collateral text, like manual annotations or speech transcripts. A third source of information is however often overlooked, namely the *surface features*. Surface features are those properties of (multimedia) documents that do not describe their content. Examples include, the length of a document, a reference to where the document is located, and the production date of a document. Although these features do not directly relate to the document's content, they can be valuable additional sources of information in a retrieval setting. In text retrieval for example, the length of a document is often used as an indicator of relevance (longer documents are more likely to be relevant). Similarly, the number of hyperlinks pointing to a document is an indicator of the importance of a document [1, 2]. Also, in the design of video browsing interfaces, the importance of surface features, like the temporal structure of video, is well-known, see for example [3]. In video search systems however, surface features are mostly ignored.

This paper assesses the usefulness of a number of surface features for multimedia retrieval. We take the TRECVID2004 collection, a collection of CNN and ABC news broadcasts from 1998, as a case study. The rest of this paper is organised as follows. The next section discusses briefly the collection and its surface features. Section 3 studies the distribution of surface features in relevant documents. Section 4 discusses how to acquire information about these distributions in practise. Section 5 shows how the knowledge about surface feature distributions can be used in a retrieval setting. We conclude the paper with a discussion of experimental results.

2 Surface Features in the TRECVID Collection

TRECVID is a workshop series with the goal of promoting progress in content-based retrieval from digital video via open, metrics-based evaluation. This paper focuses on TRECVID's search task, defined as follows:

Given the search test collection, a multimedia statement of an information need (topic), and the common shot boundary reference for the search test collection, return a ranked list of at most 1000 common reference shots from the test collection, which best satisfy the need.

The TRECVID2004 test collection consists of 70 hours of ABC and CNN news broadcasts from 1998. The collection is shot segmented and comes with a pre-defined set of keyframes. The 25 topics in the test collection are multimedia descriptions of an information need, consisting of a textual description and one or more image or video examples. For each topic, relevance judgements are available; these indicate which shots are relevant for the topic.

From the metadata associated to videos and shots, the following surface features can be extracted: the broadcaster, the date of broadcast, the time of the shot within the video, and the duration of the shot.

The TRECVID workshop prohibits exploiting the knowledge that the broadcasts in the collection are from the second half of 1998. This means for example that we cannot directly infer that if someone is looking for shots of Bill Clinton, it would be helpful to include the term *impeachment* in the query. The rationale is that this would be unrealistic. We think however it *is* realistic for a user to have some knowledge of the collection they are searching. It is indeed questionable whether such knowledge is available at system development time, since that would mean that a separate system has to be built for each new data set. Still, the information could be deduced from co-occurrence patterns in known relevant documents that are obtained through (blind) relevance feedback (Section 4).

3 Analysis of the Distribution of Surface Features

We start with an analysis of the various surface features and their distribution in relevant documents and in the collection as a whole. The statistics reported in this section are based on knowledge of the full set of relevant documents.

3.1 Local Clustering of Relevant Shots

A first observation is that relevant shots tend to cluster: when a shot is relevant for a given topic, it is likely that its neighbouring shots are relevant as well. An explanation for this is that the news broadcasts are organised in stories. A story typically shows multiple shots related to the same subject. Thus, when a topic is directly related to a news event, it is obvious that all, or at least many, shots from the story are relevant (see Figure 1). Of course, in a visual retrieval task, topics do not always ask for news events, but even there the relevant shots



Fig. 1. *Floods*; relevant images are directly related to a news event (five consecutive shots are shown)



Fig. 2. *Umbrellas*; relevant images cluster with news events (five consecutive shots are shown)

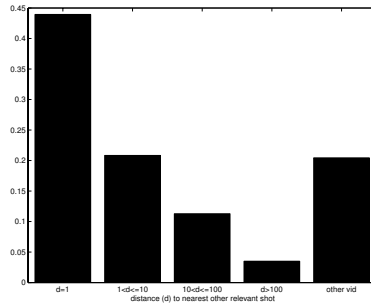


Fig. 3. Histogram of distance to nearest other relevant shot over all relevant shots. Distance measured in number of shots.

tend to cluster with one or more news event, because news stories are typically sequences of (alternating) shots in a given location or situation. When one of the shots shows a relevant item, it is likely to re-appear in other shots of the same story. For example, a news story that happens to be shot on a rainy day, is probably a good source of information when one is searching for shots with umbrellas (see Figure 2).

The distribution of the distance from a relevant shot to the next (Figure 3) shows that for almost half of the relevant shots, at least one of the neighbouring shots is also relevant. The histogram shows the average over all TRECVID2004 topics, for some topics up to 80% of the relevant shots have a relevant neighbour. Clearly, relevant shots tend to cluster.

It is debatable whether these clusters should be treated as separate relevant items. When the information need is related to a news subject, perhaps the shots should be grouped, and a sequence of related items should be presented as a single news story. However, when the information need is visual, and shots are judged on their appearance, each of them can be treated as a separate result.

3.2 Video Specific Features

Another source of information is the metadata associated with the videos, like the date of broadcast or the broadcaster. Note that these features relate to a whole video rather than to individual shots, thus, they can never distinguish between shots from the same video. Still, they may give information as to where in the collection to search for a given topic.

We studied the date of the broadcasts and found that for some topics relevant shots—or rather, broadcasts containing relevant shots—cluster in time. This is mainly the case when the topics are directly related to news events, like for the examples shown in Figure 4: *floods* mainly occurred in late October, early November 1998; Henry Hyde was the lead house manager in the Clinton impeachment trial that was televised a lot in late December 1998. We also studied the distribution over the different days of the week. For this collection and this set of topics it appears to be random, but one could imagine topics and collections where also the day of the week of a broadcast can be a valuable source of information. For example, there may be more sports news during the weekend, and film related news may cluster on Thursdays or Fridays, when new films are opening.

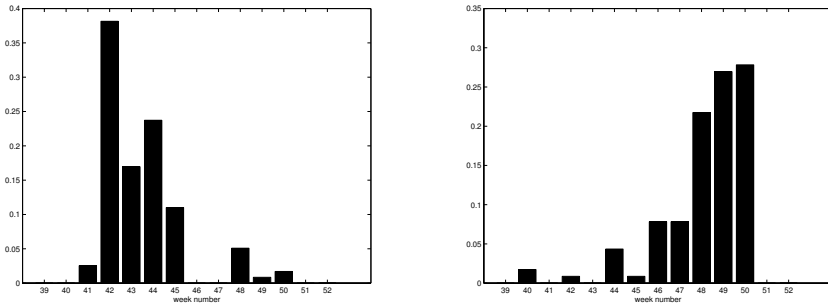


Fig. 4. Distribution of relevant shots over time for two topics: *floods* (left) and *Henry Hyde* (right)

A second attribute associated to the videos is its broadcaster. For many topics the distribution of relevant shots over the two broadcasters in the collection (ABC and CNN) is not uniform. An obvious example is the topic asking for *Sam Donaldson*, a reporter for ABC, and therefore exclusively found in ABC broadcasts. Other topics show perhaps less obvious differences. For example, relevant shots for sports related topics (*hockey rinks* and *golf*) are mainly found in CNN videos, while the topics *Saddam Hussein* or *buildings on fire* have the majority of relevant shots in ABC videos.

3.3 Time Within Video

The minute at which a shot starts may also be an indicator of relevance for a given topic. Figure 5 shows the distribution of starting minute over shots in relevant documents compared to its distribution in the collection. On average,

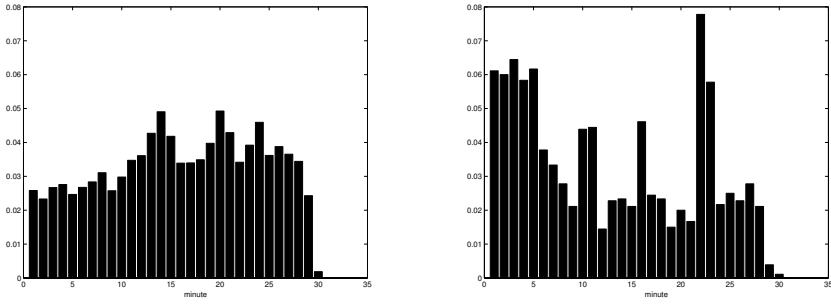


Fig. 5. Distribution of shots over minutes in collection (left) and in set of relevant documents (right)

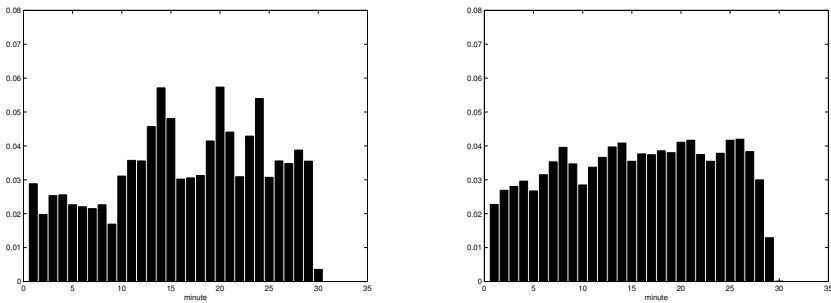


Fig. 6. Distribution of shots over minutes in CNN videos (left) and in ABC videos (right)

relevant information seems to appear more at the beginning of videos (again this is probably because of the news oriented nature of some topics). It may be somewhat surprising that shots in the collection are not uniformly distributed over the minutes. In the first 10 minutes of the videos fewer shots start than in the minutes after that. This indicates that the early shots in the broadcasts are longer. The peaks around 14, 20 and 24 minutes correspond to commercial breaks, typically composed of many very short shots. These commercial break peaks are even more pronounced in the distribution for CNN only (Figure 6). The more uniform distribution in the ABC videos can be explained from the fact that on average the shots in news ABC are slightly shorter (5.7 seconds vs. 6.6 seconds in CNN videos).

From Figure 5 we learn that relevant shots in general are likely to appear early in the videos. When we differentiate for individual topics, we may find very different patterns though. For example, the distribution of relevant shots for topics directly related to major news events (*floods*, *Clinton in front of US flag*) have an even higher peak at the beginning of videos (Figure 7). Relevant shots for sports related topics are found between minutes 20 and 25 (Figure 8).

We also analysed the duration of shots, but there seem to be no topics for which the duration of relevant shots clearly differs from the duration of non-relevant shots.

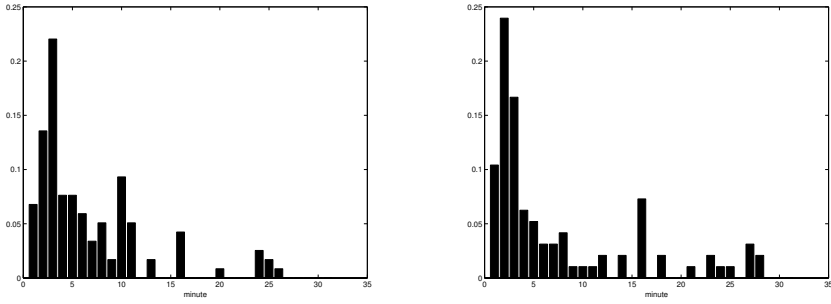


Fig. 7. Distribution over minutes of shots relevant to *floods* (left) and of shots relevant to *Clinton with US flag*

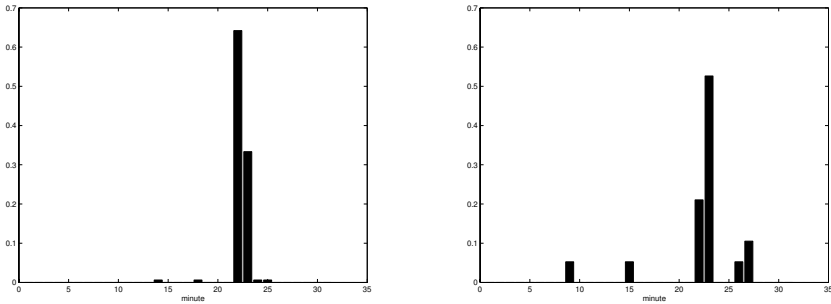


Fig. 8. Distribution over minutes of shots relevant to *hockey rinks* (left) and of shots relevant to *golf score*

4 Acquiring Surface Feature Statistics

In the previous section, we saw how surface features are potentially useful in search. Their distribution distinguishes relevant shots from non-relevant shots (especially when the features are analysed on a topic-by-topic basis). This section discusses how to obtain information about the surface feature distributions for a given topic.

4.1 User-Defined Priors

Since the user is likely to have some intuition about where relevant items appear, a simple approach would be to let the user explicitly define the surface features that are expected on relevant documents. For example, a user familiar with the data set should be able to tell the system that relevant items for sports related topics mainly appear after some 20 minutes in the news broadcast.

4.2 Topic Classification

An alternative would be to develop a taxonomy of topic types and their corresponding surface feature distributions. Such a taxonomy could for example

include separate classes for sports, politics and disasters; in addition, it could include knowledge about the broadcasters, like lists of reporters. Topics could be classified manually by the user, who no longer needs to know the relevant feature distributions as was required in the previous subsection. Alternatively, the class could be deduced automatically from the topic description using thesauri like WordNet. The corresponding surface feature distributions could be used to boost the scores of shots with the preferred features.

Some prior work exists on classifying topics and treating each topic differently [4, 5], there the classification is used to decide how to combine the various retrieval modalities (e.g., ASR, visual). An approach to use a surface feature, the time within the video, for searching for topics classified as *weather news* or *sports event* is discussed in [6]. More research is needed to investigate how well the approach extends to other topic types.

4.3 Relevance Feedback

The approaches discussed in the previous subsections only work when the topic is clearly related to a news item. One cannot expect users —let alone systems— to be able to guess which major news events coincide with shots of say *umbrellas* or *wheelchairs*. For such topics, the direct modelling of topic classes will not work.

An alternative method uses relevance feedback, and deduces the desired surface feature statistics from shots that are known to be relevant. This information can be obtained from the user who judges the top N documents of an initial retrieval step. Alternatively, we could use blind feedback and assume all of the top K documents are relevant. The distribution of the surface features in the set of relevant documents can be analysed, and shots with similar distributions can be preferred in the next retrieval round.

Blind feedback may be problematic in a visual retrieval setting, since the effectiveness of present content-based image retrieval systems is limited. Assuming that the top retrieved documents are all relevant is risky. Alternatively, this feedback step could be performed on a comparable text corpus (e.g., news paper articles from the same period). Correspondence of query terms to major news events or clustering of relevant documents in time can be found in this text corpus and transferred to the multimedia collection to improve retrieval there (the same technique has been applied to query expansion using Google news [5]). Admittedly, searching in textual corpora limits the exploitation of surface features to news related topics, because it is unlikely that visual characteristics that happen to coincide with the news event (e.g., it is a rainy day; people carry umbrellas) are mentioned in the papers.

5 Experiments

On their own, surface features may be of little use, but they may improve results obtained from traditional text-based or content-based retrieval methods. The basic idea is to run an ordinary retrieval system based on textual or visual features and then update the scores based on the surface features. Shots with

surface features that are likely to be relevant for a given topic will be pushed up the ranked list. This section discusses preliminary experiments with this approach. The experiments reported on here start from a text based retrieval system, but could easily be extended to a content-based setting.

We use a language modelling approach to information retrieval [7, 8]. This retrieval model allows for easy integration of information gathered from surface features. All that is needed is an estimate of the prior probability of relevance given a particular surface feature (or a set of such features). The content scores can then easily be updated, simply by multiplying them with this prior (similar techniques in web retrieval and XML retrieval are discussed in [9, 10] respectively). Note that in other retrieval models, surface features can be incorporated in a similar fashion by using a weighting of returned element scores based on their surface features.

All experiments described in this section are based on text only runs. The shots in the collection are described by the ASR transcripts provided by LIMSI [11]. The score of a shot given a query $Q = \{q_1, q_2, \dots, q_n\}$ is calculated as a mixture of the language models for shot, scene, video and collection, where scenes are defined as sequences of 5 consecutive shots.

$$\text{score}(\text{shot}|Q) = P(\text{shot}) \cdot$$

$$\prod_{i=1}^n [\alpha P(q_i|\text{shot}) + \beta P(q_i|\text{scene}) + \gamma P(q_i|\text{video}) + \delta P(q_i|\text{collection})], \quad (1)$$

where $\alpha + \beta + \gamma + \delta = 1$, and $P(\text{shot})$ is the prior probability of the shot.¹ All experimental runs are compared to a baseline that uses a uniform prior (i.e., $P(\text{shot}) = \frac{1}{N_{\text{shots}}}$, where N_{shots} is the total number of shots in the collection).

5.1 Retrospective Experiments

To test whether the surface features could improve retrieval effectiveness if we would have full knowledge of their distributions, we experimented with estimating the distributions from the full relevance judgements. Of course, in a realistic setting, relevance information will never be fully available, but retrospective analysis allows us to explore the potential of the surface features.

The length prior is a linear function of the number of words in the shot's transcript, as is common in the language modelling approach to information retrieval. All other priors are based on the empirical distribution of the surface features in the relevant set. We are interested in the probability of relevance given a surface feature (sf), which can be estimated based on the distribution of the feature in the relevant set and in the collection as follows:

$$P(\text{rel}|sf) = \frac{P(\text{sf}|\text{rel})P(\text{rel})}{P(\text{sf})} \propto \frac{P(\text{sf}|\text{rel})}{P(\text{sf})} = \frac{\#(\text{rel}, \text{sf})}{\#(\text{sf})}, \quad (2)$$

¹ We use the mixing parameters optimised on the TRECVID2003 test set: $\alpha = 0.4, \beta = 0.4, \gamma = 0.02, \delta = 0.18$.

Table 1. Mean average precision (MAP) for various priors estimated on full relevance judgements (retrospective)

run name	description	MAP
baseline	no prior	0.075
length prior	size of transcript in words	0.075
source prior	broadcaster (ABC or CNN)	0.081
week prior	week of the year of broadcast	0.087
duration prior	duration of shot in seconds	0.095
minute prior	start of shot in minutes from start of broad- cast	0.096

where $\#(\text{rel}, \text{sf})$ is defined as the number of relevant documents with surface feature sf , and $\#(\text{sf})$ as the total number of documents with that feature. Table 1 lists the priors studied and reports the mean average precision for each.

The table shows that all studied surface features could potentially improve over the baseline, except for document length. The limited influence of a length prior (especially when compared to its importance in e.g. text retrieval) could be explained from the fact that the shots do not vary much in length. In addition, unlike the other priors, the length prior is not topic-specific. Estimating the length prior on a topic-by-topic basis could perhaps give some improvement, but a large effect is unlikely given the flat distribution of shot lengths, and overfitting is likely.

5.2 Feedback Experiments

After learning that the surface features can potentially improve retrieval results, we experimented in a more realistic setting, one where no full knowledge of the relevant set is available. We took the relevance judgements of the top N results of the baseline run to estimate the surface feature priors from, thus mimicking relevance feedback on the top N retrieved documents. Based on these priors, we produced a new ranking of shots keeping the top N fixed.² The new ranking is compared to the baseline using mean average precision. Computing the priors directly using Equation 2 from the small amount of data available in the top N would result in poor estimates due to over-fitting. To avoid this, we interpolate the estimates obtained from the top N with a uniform prior:

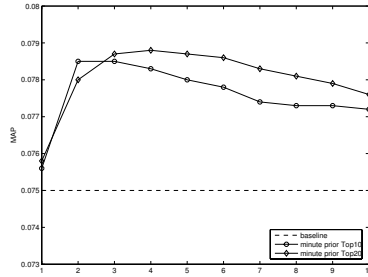
$$P(\text{shot}) = \lambda P(\text{rel}|\text{sf}) + (1 - \lambda) \frac{1}{N_{\text{shots}}}.$$

This way shots with surface features that are not observed in the relevant documents in the top N still have a chance of being retrieved. We experimented with feedback on the top 20 results ($N = 20$), and with various values of the mixing parameter λ . Table 2 shows the results.

² Re-ordering the top N is likely to produce a higher MAP, but does not give the user new results.

Table 2. Mean average precision (MAP) for priors estimated on Top 20 feedback with different values for the mixing parameter λ

run name	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.8$	$\lambda = 1.0$
baseline	0.075	0.075	0.075	0.075
source prior	0.068	0.068	0.069	0.073
week prior	0.059	0.059	0.059	0.060
duration prior	0.046	0.046	0.046	0.047
minute prior	0.055	0.055	0.056	0.057

**Fig. 9.** Mean average precision for minute prior based on Gaussian kernel density estimation

The mean average precision scores show that, despite the potential, none of the surface feature priors is of practical use in this setup. One of the reasons could be that we are over-fitting on the relevant documents found in the top N . The granularity of measuring some of the features may be too small. Take for example the minute prior, where we look at the starting minute of a shot. Once we have found a relevant shot that starts at a certain minute, we may decide to prefer shots around that time rather than only shots that start at the exact same minute as we have done so far. To investigate this, we take a closer look at the minute prior, and follow the approach of Lin and Hauptmann [6]. They use kernel density estimation to estimate the density of specific classes of video (sporting events, weather news) over time. Like Lin and Hauptmann, we use Gaussian kernels, but in our case no classification of the topic is needed; we estimate on a topic-by-topic basis. The width of the kernels (σ) is varied from 1 to 10 minutes. The resulting MAPs are shown in Figure 9. Using these smooth minute priors estimated on either top 10 or top 20 feedback, yields an improvement over the baseline. Taking a closer look at the results reveals that the improvement is due to a few topics only (related to major news events and sports events). Nevertheless, it does not harm the other topics. It is interesting to see that prior information can be obtained from such a small amount of training data, and that it can be used in a practical situation.

6 Discussion

This paper has shown that surface features can contain useful information for multimedia retrieval. Knowledge about the relevant features for a given topic can

be used to narrow down the search space, or to prioritise documents with certain characteristics. Retrospective experiments with full knowledge of the relevance judgements have shown the potential of using such features. A more detailed study of one of these features, the time of shots within a broadcast, showed that also in a practical situation, surface features can be useful.

The effect of the minute prior may be partially attributed to retrieving the neighbours of known relevant shots. In a general retrieval task, this may not be that interesting, and a sequence of related shots should perhaps be treated as a single retrieval unit. But in the retrieval task studied here, we are searching for shots because of their visual appearance. In such a setting, each shot can be seen as a separate relevant document, since each represents a different view of the same event.

Although the features studied in the present paper and their effectiveness may be restricted to the news broadcast domain, the principle is generally applicable. As long as a collection is relatively homogeneous, useful surface features are likely to exist.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30** (1998) 107–117
2. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46** (1999) 604–632
3. Lee, H., Smeaton, A.F.: Designing the user interface for the Físchlár digital video library. *Journal of Digital Information* **2** (2002)
4. Yan, R., Yang, J., Hauptmann, A.G.: Learning query-class dependent weights in automatic video retrieval. In: *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, ACM Press (2004) 548–555
5. Chua, T.S., Neo, S.Y., Li, K.Y., Wang, G., Shi, R., Zhao, M., Xu, H.: Trecvid 2004 search and feature extraction task by NUS PRIS. In: *TREC Video Retrieval Evaluation Online Proceedings*. (2004)
6. Lin, W.H., Hauptmann, A.: Modelling timing features in broadcast news video classification. In: *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*, Taipei, Taiwan (2004) 27–30
7. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (1998) 275–281
8. Hiemstra, D.: A linguistically motivated probabilistic model of information retrieval. In Nicolaou, C., Stephanidis, C., eds.: *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*. Volume 513 of *Lecture Notes in Computer Science*, Springer-Verlag (1998) 569–584
9. Kraaij, W., Westerveld, T., Hiemstra, D.: The importance of prior probabilities for entry page search. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press (2002) 27–34
10. Ramirez, G., Westerveld, T., de Vries, A.P.: Structural features in content oriented XML retrieval. Technical Report INS-E0508, CWI, Amsterdam, The Netherlands (2005)
11. Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI broadcast news transcription system. *Speech Communication* **37** (2002) 89–108

Toward Consistent Evaluation of Relevance Feedback Approaches in Multimedia Retrieval

Xiangyu Jin¹, James French¹, and Jonathan Michel²

¹ University of Virginia, Charlottesville VA 22903, USA

² Science Applications International Corporation, Charlottesville VA 22911, USA

Abstract. Many different communities have conducted research on the efficacy of relevance feedback in multimedia information systems. Unlike text IR, performance evaluation of multimedia IR systems tends to conform to the accepted standards of the community within which the work is conducted. This leads to idiosyncratic performance evaluations and hampers the ability to compare different techniques fairly. In this paper we discuss some of the shortcomings of existing multimedia IR system performance evaluations. We propose a common framework in which to discuss the differing techniques proposed for relevance feedback and we develop a strategy for fairly comparing the relative performance of the techniques.

1 Introduction

The information retrieval (IR) task is to find documents relevant to a searcher's information need. Relevance feedback may be offered by an IR system as a mechanism for more effectively capturing a user's information need. Usually relevance feedback is carried out by having the user select positive/negative examples from the current retrieval result to be used to modify the query for the next retrieval iteration.

Research in relevance feedback for multimedia retrieval is being carried out by several different disciplines, for example, computer vision [1, 2], database management [3, 4, 5], information retrieval [6, 7], human computer interaction [8], artificial intelligence, and even psychology. These groups tend to follow different traditions within their communities and, in particular, they often have different standards and approaches to evaluation. Unlike text IR, which is an established area with general agreement upon evaluation protocols such as TREC (Text Retrieval Conference),¹ relevance feedback research in multimedia retrieval still does not have a generally accepted evaluation methodology.

Take content-based image retrieval (CBIR) as an application example. On one hand, different research groups tend to use different image libraries. In the early days of CBIR research, many groups used the COREL² image collections[1, 9, 4]. Unfortunately, as pointed out by Müller[10], they tend to use different subsets which makes cross system comparison impossible. More recently, research groups have begun using the test collections of TRECVID³ [11, 12]. Even if the

¹ <http://trec.nist.gov/>

² <http://www.corel.com/>

³ <http://www-nlpir.nist.gov/projects/trecvid/>

same image collection is used, a different groundtruth can be used for evaluation purposes. Manual judgment (using human labeling[9, 4]), automatic judgment (using a reference retrieval system [1, 2]), and semi-automatic judgment (using system-assisted human labeling [13]) can be used as groundtruth. These different “correct answers” make it extremely difficult to perform cross system comparisons. On the other hand, relevance feedback tests can be executed by machine simulation according to a pre-defined groundtruth (system-oriented) or by real users via some interface (user-oriented). For example, in TRECVID each site can submit its runs based on any feedback approach they executed and usually this process involves human users. However, the user-oriented feedback test methodology is not ideal for cross system comparison purposes since this usually requires a large amount of human effort and a human judge is not as consistent as a machine is (i.e., human user’s experience can vary from person to person and time to time). These issues make cross system comparison almost impossible since too many factors could affect the feedback performance other than the feedback algorithm itself. Only very recently have testbeds which can provide stable human judgment and methods to balance human bias appeared. One such approach is the clarification forms in the TREC HARD track.⁴

In addition to these problems, rank normalization [7], which is considered almost a standard process in text IR, is seldom paid enough attention to in multimedia retrieval. Most current research work in this area neglects this problem and does not perform proper rank normalization when comparing relative performances. These problems make it hard for us to study the relations among different feedback approaches and hard to fairly compare competing techniques under a realistic application environment.

We make two contributions in this paper. First, we briefly summarize the relevance feedback approaches in multimedia retrieval by putting them in a common framework so that each approach can be treated as a special case and their intrinsic relations can be studied. Second, we point out the evaluation problems in previous research and demonstrate how we perform a fair comparison for three typical feedback approaches in large scale test beds (both text and image) which are indicative of real application. We also show that improper evaluation methodology can lead to quite different and even contradictory conclusions.

The rest of the paper is organized as follows. Section 2 provides an overview of several major approaches to relevance feedback in multimedia retrieval. In section 3 we point out several problems in their evaluation. In section 4 we describe our framework and how we perform a fair comparison among these approaches. Sections 5 and 6 discuss our experiments.

2 Relevance Feedback in Multimedia Retrieval

2.1 Overview

Approaches for relevance feedback in multimedia retrieval often involve a retrieval procedure to handle multiple query points, i.e., multi-query retrieval.

⁴ <http://trec.nist.gov/tracks.html>

Many researchers working on multi-query retrieval do not directly connect them to relevance feedback. However, there is strong relation between these two notions. If we are given a solution to multi-query retrieval, we can implement a corresponding relevance feedback process, and vice versa. The process of relevance feedback can be abstracted as the following steps:

1. An initial query set S gives rise to an initial retrieval result L
2. If the termination condition is met then END
3. Select new feedback examples from L and put them into S
4. Issue a multi-query retrieval and get a new retrieval result L' , let $L = L'$
5. Goto step 2

Our abstraction of the relevance feedback process is a little bit different from the traditional way. There is no independent “refinement” step. The refinement task is seamlessly carried out by the multi-query retrieval in step 4. This is achieved by issuing a new multi-query search whose query both includes the old query points and the new feedback ones. Therefore, in the following, we focus on the solution to multi-query retrieval. We note that our notion is not the same as a multi-modal query. While our queries could be multi-modal, there is no requirement that they be.

In order to simplify our discussion, we restrict our topic to positive feedback examples. We do this for two reasons. First, approaches can be easily extended to negative examples. Second, negative examples may be harder to select in a real user interface since users may find it harder to judge the extent of dissimilarity than similarity.

In a distance based IR system, both the documents and queries can be abstracted as points in some space referred to as document points and query points in this paper. Each pair of points’ distance is defined by some distance function D . The process of retrieval can be interpreted as getting the document points in some neighborhood of the query points. When there is only a single query point to consider, the query q to each document d ’s distance is evaluated by $D(q, d)$. The retrieval is a nearest neighbor search process. When multiple query points are provided, we must extend the distance function D so that it can handle the distance between a query set Q and a document d . There are two possible solutions to construct such D' : combine queries and combine distances.

Combine Queries Approaches. The combine queries approach tries to synthesize a single query point from the given query point set and put the synthetic query into the distance calculation. The idea is to create a mapping to map a set of query points $Q = \{q_1, q_2, \dots, q_T\}$ to a single query q and define $D'(Q, d) = D(q, d)$. The query q can be created either by selecting a representative from Q or generating a point which may be not in Q . The latter approach is usually applied when the space is a vector space, thus the new query point is usually generated via a linear combination of all points in Q :

$$q = \sum_{i=1}^T w_i * q_i \tag{1}$$

Where w_i is the corresponding weight for q_i . In this case, q must reside in the convex hull of Q . If all queries are regarded as having the same importance, q is actually the center of Q 's convex hull (referred to as query-center in this paper). Query-point-movement [6], re-weighting [1], and MindReader [5] all follow this track. They all use a synthetic single query point to retrieve the data. Figure 1(a) gives an example of using the query center. The query set Q is composed of four queries q_1, q_2, q_3, q_4 . The combine queries approach first selects a point q (the query center of Q) and uses the distance from q to d as the expected distance from Q to d .

Combine Distances Approaches. The combine distances approach uses an aggregate function [14] to combine the set of distances (each query point's distance to the document point) to a synthetic one. A T -ary aggregate function φ should satisfy: $\varphi(x_1, x_2, \dots, x_T) \leq \varphi(x_{1'}, x_{2'}, \dots, x_{T'})$ if $\forall i(x_i \leq x_{i'})$. The power mean is usually employed for distance combination. A power mean is a generalized mathematical mean for a data set and is defined over a T -element set $X = \{x_i | 1 \leq i \leq T\}$ as: $mean_\alpha = (1/T \sum_{i=1}^T x_i^\alpha)^{\frac{1}{\alpha}}$. The new distance function D' can be regarded as a weighted power mean of distances from individual query points to d :

$$D'(Q, d) = \left[\sum_{i=1}^T w_i * D(q_i, d)^\alpha \right]^{\frac{1}{\alpha}} \tag{2}$$

We further define $D'(Q, d) = 0$ if both $\alpha < 0$ and $\exists i(D(q_i, d) = 0)$ and thus avoid possible division by 0. This is a reasonable modification since points in Q would have 0 distance to themselves. Moreover, in a real implementation, if the combined distance is only used for ranking purpose, we can omit the last power operation and just simply sort in reverse order if $\alpha < 0$. Query-expansion in MARS [2] (not the same notion in text retrieval), FALCON [3], and QCluster [4] follow this track. Figure 1(b) shows an example of using arithmetic mean as the aggregate function. Instead of using a query center, the average of distances from all query points to d is now regarded as the distance from Q to d .

In the next section we describe some of the significant research efforts for relevance feedback in multimedia retrieval.

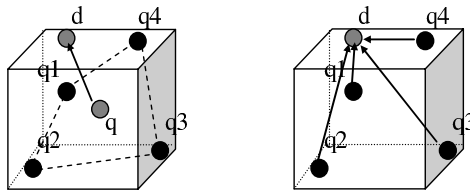


Fig. 1. Combine queries VS combine distances. q_1, q_2, q_3, q_4 are four queries in Q , d is a document.

2.2 Combine Queries Feedback

The intuition for combining queries is as follows: if the user-interested documents are clustered in the space, we could find the relevant documents by an “ideal” query (the cluster center). In this case, the feedback process can be abstracted as shifting and reshaping a query region in the space to fit the user-interested area. The two tasks are referred to as query-point-movement and reshaping in this paper. The first task aims at shifting the query region toward a proper location and the latter one tries to stretch the query region to fit the user interested area optimally.

The classical query-point-movement is expressed by Rocchio’s formula [6], which is based on the vector space model. The new query point is the previous query point plus some movement toward positive queries and some movement away from negative queries. Therefore query-point-movement provides continuous movement of the query point in the space, thus shifting the location of the query region to a proper place. In our framework, Rocchio method’s implementation as multi-query retrieval is to use a linear combination to combine the queries as in Eqn. (1).

Reshaping approaches all have some assumption on the distance function. Most current research requires that the distance function be a weighted L_p metric, e.g., MindReader requires squared Euclidian distance. Suppose each document is represented by an M -dimensional feature vector (if multiple features are employed, just concatenate them to be a single long vector), all dimensions are normalized in the same scale $0-1$. Define the general squared Euclidian distance function to be

$$D(q, d) = [q - d]^T P [q - d] \quad (3)$$

where P is called the distance matrix, which is an $M \times M$ symmetric matrix satisfying $\det(P) = 1$. The query region is a hyper-ellipse in the space. Modifying the distance matrix P will change the shape of the hyper-ellipse. Therefore, reshaping can be interpreted as a process to refine the distance matrix.

The classical Rocchio method shifts the location of the query region but does not change the shape of it. Therefore it keeps P as an identity matrix. This may not capture the user’s information need well. Suppose we have 2D database composed of people’s heights and weights. If we want to find people whose weight is around 140 pounds, the user interested region is actually a sharp band along the line $weight = 140$. But Rocchio method always forms a circular query region which cannot fit the intended area. In order to remedy this problem, a natural thought is to allow the query region to be an ellipse. Rui *et al.*[1] proposed a standard deviation approach to adjust the distance matrix in their retrieval system MARS. In this work, they assume the distance matrix to be a diagonal matrix (the standard deviation approach is not restricted to squared Euclidian distance). The data on the diagonal are the weights for different feature components. The basic idea is the higher the variance of queries points along an axis, the lower is the importance of this dimension. So the corresponding weight for this dimension should be low. Therefore, the query region is now stretched to an ellipse.

The re-weighting approach regards each dimension to be independent and orthogonal. Therefore, the resulting hyper-ellipse should be aligned to the axes. However, there may exist queries that actually infer some relation between axes. For example, the “diagonal query”, which is presented in [5], infers a linear relation between axes. Still consider the previous 2D weight-height database as an example. If we want to find “good shape” people, the expected region is a band along the diagonal, where weight/height varies within a range. In this case, re-weighting cannot form an optimal ellipse since neither aligns to weight axis nor height axis. In the MindReader system, Ishikawa et al. loosens the restriction on P further (P is a symmetric matrix) so that the hyper-ellipse can be arbitrarily shaped in space.

2.3 Combine Distances Feedback

Combine query approaches all hold strong assumptions which are hard to achieve in operational systems. For example, the Rocchio method requires the space is vector and reshaping requires that the squared Euclidian distance function is used. To avoid these restrictions, combine distances approaches are deeply explored recently.

The idea of combining the distances of a document to multiple query points can be traced to combining multiple evidence in text retrieval. Several investigators have explored improving retrieval effectiveness by combining multiple information sources, such as different search strategies [15] and different query representations [16]. These combining techniques (referred to as fusion, merging) can result in improved retrieval effectiveness. In our early work [9] we showed if we treat each query point’s retrieval result as an independent information source, the fused result can be regarded as the refined search result. Therefore, we can introduce techniques which proved to be effective in text fusion into multi-query retrieval. For example, we can use rank merge instead of raw similarity merge to avoid the complex normalization issue. We can also truncate each query point’s retrieval result to improve merge efficiency. In [17], some merging techniques for text retrieval are summarized. The most commonly used are COMBSUM, COMBMIN and COMBMAX which are just special forms of the aggregate function in Eqn. (2) where $\alpha = 1$, $\alpha \rightarrow -\infty$, and $\alpha \rightarrow +\infty$ respectively.

The early work to apply combine distances in multimedia retrieval is MARS’s query expansion [2]. Apart from the difference in application, the algorithm of MARS’s query expansion is exactly the same as the COMBSUM approach in Shaw’s paper [17]. In [2], a weighted arithmetic mean (where $\alpha = 1$ in Eqn. (2)) to combine the distance. They claim their approach can adapt to any arbitrary shape in space, such as an triangular query region. Hence, it is superior to query-point-movement.

Wu et al. [3] proposes a much more sophisticated approach for merging the distance where a general power mean is used (c.f. Eqn. (2)). They further distinguish the situations of $\alpha > 0$ and $\alpha < 0$ as fuzzy AND merge and fuzzy OR merge, respectively. For fuzzy AND merge, if a document is judged as dissimilar by any query point, then the document is regarded as dissimilar. On the

contrary, for fuzzy OR merge, if a document is judged as similar by any query point, then the document is regarded as similar. These two categories of merging algorithm behave differently in forming the query region. For fuzzy AND merge, any retrieved document should be close enough to all the query points. If the distance function is metric by triangle inequality this ensures the retrieved documents have small pair-wise distances, i.e., they are clustered in the space. But for fuzzy OR merge, this property doesn't hold. This means generally in a metric space fuzzy AND merge tends to form a continuous region which covers the central region of query points but the fuzzy OR merge can form several disjoint regions. Applying power mean in distance measure is not new to IR. A similar idea can be traced to the extended Boolean IR approach proposed by Salton et al. [18], where a p -norm model is proposed to make fuzzy judgment toward the degree of document's matching to a query. Other similar ideas can be found in Korfhage's distance metrics [19].

QCluster [4] is a recent work that tries to unite the merit of combine queries and combine distances approaches. First, the feedback documents are clustered into several groups by some clustering algorithm. Then a combine queries approach is applied within each group, so that each group has its query center and an optimal distance function. MindReader [5] is employed for this purpose. These query centers form a new query set Q , and each query q_i has an associated distance matrix p_i with it. Finally, combine distances approach is applied on query set Q to retrieve documents. A query point q_i in Q will evaluate the distance to a document according to its own distance function. From this viewpoint, it is a mixture of combine queries and distances approaches. The major concern of QCluster is to separate intra-cluster and inter-cluster feedback processes, where combine queries approach is applied to intra-cluster feedback and combine distances approach is applied to inter-cluster feedback. This is a very complex approach whose reliability highly depends on the clustering effect.

3 Evaluation Problems

Performance evaluation is critical for properly understanding relevance feedback research in multimedia retrieval. However, as we have already noted, the fragmented nature of the research effort across different domain communities leads to evaluations which cannot be compared to one another. Unfortunately, we seldom see continuous evaluation work to test these ideas under a practical environment either by the same group or others. This makes it extremely difficult for us to assess their usability for real applications. We now list several common evaluation problems that often appear in the published literature for multimedia evaluation.

Problems with the dataset. This is the most commonly occurring problem and appears in the following guises.

1. An algorithm is proposed based on some assumption. Instead of verifying this assumption on real application data, the algorithm is evaluated against an arbitrarily constructed dataset where the assumption already holds. But the assumption is still left unverified.

2. The experiment is performed on a real dataset, but the dataset is small, low dimensional, or highly structured and is far from the real application environment. Under this “real” dataset, the proposed approach will work well but it is still an open question whether it would work well with a much larger, higher dimensional, and unstructured dataset.
3. The dataset itself has no problem, but the groundtruth is arbitrarily generated and is biased in favor of the proposed approach. This is the most complicated case facing the reader and it is usually hard to find out what is really going on.

For example, MindReader [5] is based on the assumption that “diagonal” queries exist. However, instead of verify this assumption, they perform a comparison between their approach and the MARS re-weighting approach in arbitrarily constructed datasets and a real Montgomery Country dataset. The latter one is very small and a structural 2D map, where points near I-270 are of interest, i.e., a diagonal query exists. The experiments cannot be used to prove MindReader will be more useful than pure query-point-movement in a real multimedia retrieval environment since we do not know whether such diagonal queries would exist. Actually, the diagonal query does not have an intuitive interpretation in a multimedia retrieval environment.

Another example is the MARS re-weighting [1] approach. Re-weighting is based on the assumption that the “ellipse” query is more suitable for a user’s information need than “sphere” query in a multimedia retrieval application. However, instead of verifying this point, [1] generates the groundtruth by arbitrarily constructed hyper-ellipse in the feature space ⁵ and then show us that their re-weighting finally converges on the “ideal” distance matrix. However, this evaluation cannot answer the question of whether re-weighting can perform better than pure query-point-movement in a real CBIR application. [1] did not provide any comparison between these two approaches. It is unknown how much improvement (or even if there is such improvement) for the re-weighting approach in a real retrieval system. Even if the assumption is correct, we must notice multimedia retrieval systems usually employ very high dimensional features. This makes it very hard to mine the relation among hundreds of dimensions via several dozen feedback examples. It is highly possible that the noise can overwhelm the real user intention and result in a lower performance. A similar problem also exists in [2], where the groundtruth is generated via a similar technique rather than from semantic level user judgments.

Problem with the comparison. Another category of problems is relevant to comparison. We further classify them as *no comparison*, *misused comparison*, and *unfair comparison*. In the first case, the author proposes some algorithm without any comparison to others, but gives evaluation scores according to their own experiments, including convergence speed, precision-recall graph, etc. However, as noted earlier because we lack a standard evaluation test bed in this area these

⁵ This is done by constructing some arbitrary distance matrix P and then performing real image retrieval to get the retrieval result as the “ideal” result in user’s mind.

scores seldom give the reader an intuitive feeling for how well the proposed approach works. In the second case, the author proposes a new algorithm B based on some modification of an algorithm A . But instead of comparing B with A it is compared with another poorer algorithm C and concludes that B 's performance is much better. In the third case, the evaluation treats feedback examples inconsistently, thus resulting in unfair comparison. Suppose the retrieval result is in the format of a ranked list. Normalization issues arise when we compare the retrieval results directly. On one hand, when we compare different feedback approaches, some feedback approaches (such as fuzzy-OR-merge) will automatically shift those feedback examples to the head of the result list, while others (such as query-point-movement) will not. Comparing them directly is unfair because any approach can shift the feedback examples in post-processing. On the other hand, when we compare different feedback iterations, it is unclear how much of the performance improvement is contributed from the user and how much is from the retrieval system since those training samples also join the evaluation. Rank normalizations, such as "fluid" and "frozen" [7], are the techniques dealing with these problems. Without a rank normalization process, we would exaggerate the improvement of some feedback approaches. Although rank normalization is generally accepted in text IR community, it is often neglected by multimedia retrieval.

The experiments of QCluster [4] are an example of misused comparison. [4] compare the performance of QCluster approach with query-point-movement and query expansion in MARS and conclude that QCluster is the best. However, they should at least compare their work to FALCON's fuzzy-OR merge, which their approach is based on, but not any earlier approach. Therefore, it is very hard to see their contribution in this experiment. In our experiments (c.f. sec. 5 and 6), we show that the COREL database has some attributes which will make any approach similar to fuzzy-OR outperform those similar to fuzzy-AND. Therefore, it is not convincing that merit is gained via such a complex approach.

None of the work listed in section 2 performs rank normalization. The problem is more severe in the case when the relevant set is small. For example, [4] use COREL groundtruth so for each query there are 100 relevant images. This is a small number which makes the accumulated feedback examples take a large portion of the relevant set. Recall will show great improvement by any approach which shifts the feedback examples to the head. FALCON [3] is another example but the problem is less severe since the arbitrarily generated dataset has a large relevant set for each query, so the improvement in recall is relatively objective, because the number of accumulated feedback examples only takes a small portion of the relevant set.

Impractical parameter settings. Sometime a paper uses an evaluation methodology which is unlikely to appear in a real application usually by assuming the user to be "diligent." For example, [1] suppose the user would look through the top 1100 retrieval results to find feedback examples. [3] compares feedback iterations after 30 times, but we know a real user would seldom have the patience to do that. [4] and [2] assume the user would feed back all relevant images in the top

100 retrieval result. We note that this is a possible reason why most feedback contribution appears in the first iteration. Because the COREL database only has 100 relevant images for each query, feeding back all relevant images in the top 100 retrieval result may already include most of them.

4 Evaluation Framework and Rank Normalization

In the following, we describe the evaluation framework we use to perform comparisons among different relevance feedback approaches fairly, including our implementation of rank-shifting and rank-freezing. Then we give an example of evaluating three most typical feedback approaches under large scale image and text retrieval test beds. Here our purpose is not to make extensive empirical studies to compare these approaches but rather to show that evaluation methodology is so important that by itself it can lead to the appearance of performance improvement when none exists or cause one to draw erroneous and/or contradictory conclusions when comparing systems.

First, we define the following notions and operations:

1. A document list $L = (L_1, L_2, \dots)$ is a finite, non-duplicated list of document
 - (a) Define $s(L) = \{L_1, L_2, \dots\}$ is the set of elements in L
 - (b) Define $len(L) = |s(L)|$ as the length of L
 - (c) Define $rank(d, L)$ ($d \in s(L)$) return an integer indicating the rank of a document d in L
 - (d) Define $top(L, K) = (L_1, L_2, \dots, \min(len(L), K))$ as a truncation to keep top up to K elements in L
 - (e) Define $filter(L, S)$ (S is a document set) as a function to exclude elements in $s(L) \cap S$ from L
 - (f) Define $L_1 \circ L_2$ as the operation to concatenate two lists
 - (g) Define $sort(S, L)$ ($S \subseteq s(L)$) as the function to sort a set S into a list according to the order of L
 - (h) Define $insert(d, L, p)$ ($d \notin s(L)$) as an operation to insert a document d in the p -th position of a list L
2. Define relevance function $rel(d)$ as a Boolean function to show whether the document d is relevant for the current information need
3. Define feedback selection function $S' = f(L, k, S)$, which returns a relevant document set S' from $s(L) - S$, which contains upto k relevant documents, i.e., $(S' \subseteq s(L) - S) \wedge (|S'| \leq k) \wedge (\forall d(d \in S' \rightarrow rel(d)))$
4. The evaluation function $E(L)$ returns a non-negative real number reflect some measure on L , e.g., precision at 100, average non-interpolated precision.

Now we give our evaluation framework of relevance feedback process:

1. The retrieval system returns L as the initial retrieval result; $S = \{\}$
2. $L = top(L, K)$, output $E(L)$
3. if some termination condition met then END.; else $S = S \cup f(top(L, \hat{k}), k, S)$
4. Issue S as a multi-query search and get the retrieval result as L'

5. Perform rank normalization on L' and get the normalized result L''
6. $L = L''$; goto step 2

We offer two possible implementations for rank normalization: rank-shifting and rank-freezing. Rank-shifting moves all feedback examples to the top of the refined retrieval result. Rank-freezing keeps those feedback examples' ranks in the previous retrieval result unchanged in the refined retrieval result, as if they are "frozen" there. These two approaches both "normalize" the performance improvement contributed from the user feedback examples. Rank-shifting makes them equal by maximizing them. Rank-freezing makes them equal by minimizing them. We should note that these two techniques are not necessarily performance order preserving under arbitrary performance measures, although in general they differ only when performance is very similar. We can always give negative examples that would generate different order for some measure. But since the variation is very small, we still claim either technique can be used to compare feedback approaches (due to space limitation, we skip the discussion here). Rank-shifting is relatively easier to implement and rank-freezing is more objective when comparing the performance improvement over iterations. The rank normalization process in step 5 can be implemented either as:

<p>Rank-shifting</p> $\hat{L}' = filter(L', S);$ $\hat{S} = sort(S, L);$ $L'' = \hat{S} \circ \hat{L}';$	<p>or Rank-freezing</p> $\hat{L}' = filter(L', S);$ $\hat{S} = sort(S, L);$ <p>for $i = 1$ to $len(\hat{S})$ do</p> $insert(\hat{S}_i, \hat{L}', rank(\hat{S}_i, L));$
-----------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The retrieval system always outputs a truncated document list of length up to K for evaluation. Each feedback iteration adds up to k new documents to the feedback set S . Then S is issued as a multi-query search to get a "refined" retrieval result. Finally, evaluation is performed on the truncated "refined" result and if some condition is met we terminate the iterations. There are several parameters need to be decided when implementing a feedback test on a retrieval system. K is the length of the document list that the retrieval system returns. \hat{k} is the length of documents that user is supposed to look through to find relevant documents. k is the maximum number of documents that a user would like to feed back to the retrieval system. Obviously, $K \geq \hat{k} \geq k$. Another thing to be decided is the selection strategy used in feedback selection function $f(L, k, S)$, which is relevant to the user scenario.

5 Experimental Environment

5.1 Test Beds

CBIR test bed. In this test bed, 3400 color images (34 categories, each with 100 images) from the COREL database are indexed by our basic CBIR system [9]. This CBIR system is based on 7 global visual features and each image is

represented by a vector. In [9], we have shown our basic CBIR system provides a baseline which is competitive to top CBIR systems such as Simplicity [20] in the same test bed. We use our basic CBIR system because combine query approaches can be implemented.⁶ The COREL category is regarded as the ground truth, i.e., all the images in an image category are relevant to all the other images in the same category and not relevant to any other images. Six images from each category are randomly chosen as the query images, hence there are 204 queries in total. In [21] we have shown this test bed is a good representative to a much larger scale test bed (composed of 60K images) for relevance feedback experiments.

Text IR testbed. In this test bed, Lucene⁷, which is an open source text search engine offered by Apache Jakarta project, is used to index Tipster data used in TREC-3 evaluation (disks 1 and 2). The dataset consists of about 750K documents. TREC topics 151-200 are used for query formulation. These topics have been assessed against the data on disk 1 and 2 and relevance judgments are provided. Since Lucene uses a VSM (Vector Space Model) like scoring algorithm to rank the documents, each document is represented as a vector in its implementation. The data is indexed with a standard analyzer provided by Lucene. The queries are created by a user by reading all these 50 TREC topics⁸ One query for each topic. These queries are short and typically 2-5 words in length. We assign these words equal importance and form them into vector queries. This test bed is close to real queries input by a common user, hence simulating a “practical” environment.

5.2 Feedback Approaches

We implemented and compared three feedback approaches which typically represent today’s relevance feedback technology in multimedia retrieval:

Combine queries: Query-Point-Movement (QPM). In this implementation, we use the classical Rocchio method and give all query points equal importance (i.e., the initial query is weighted the same as the feedback ones). Re-shaping is not considered at this time because the distance function is not a squared Euclidian distance. The old query points and new feedback ones are averaged to get a new query point.

Combine distances: AND-merge and OR-merge. In this implementation, each feedback document is issued as a query. In the CBIR test bed, feedback images are directly fed into the CBIR system for they directly support QBE (query-by-example). In the text test bed, where QBE is not directly supported, the document’s representing vector is issued as a query. For efficiency purpose, only 20 most significant components of the representing vector are considered in

⁶ We also conducted the same experiments with Simplicity and drew similar conclusions. Due to space limitations, we do not report all the results here.

⁷ <http://jakarta.apache.org/lucene/>

⁸ We also performed the same experiments using different query sets formulated from the same topics and drew similar conclusions. Due to space limitations, we do not report all the results here.

formulating the query. Retrieval results and previous search result are merged by assigning the same weight for the initial query and each feedback example. We use mid-rank [22] merge instead of merging the raw distance output by the search engine. First, using rank can avoid complex normalization issues arising from the retrieval system itself. Second, in our experience rank merge is competitive in performance to those complex merging approaches which use the raw distances. For AND-merge we choose $\alpha = 1.0$ and for OR-merge we choose $\alpha = -1.0$. For other choices of α we have similar results.

5.3 Other Experimental Settings

We let $K = 150$ for the CBIR testbed and $K = 1000$ for the text-IR testbed. These are commonly used values for evaluation in the corresponding area. We assign $\hat{k} = 150$ for both test beds because it is a reasonable length for a user to look through to find relevant documents. We also assign $k = 8$, for we think it is a reasonable number for a user to feed back into the system. We use a sequential scan for feedback selection. That is, we suppose the user reads at most the top 150 documents and selects the first eight (actually, up to 8) new relevant documents encountered which have not yet been selected in any earlier feedback iteration.

6 Experimental Results

The evaluation metric we use is average non-interpolated precision which is a rank sensitive measure. We only compare the performance of the first three feedback iterations. The results are shown in Fig. 2. The left and right columns show the results from the CBIR and TEXT test beds, respectively. The three rows show the results without rank-normalization, with rank-shifting, and with rank-freezing, respectively.

First we compare the difference among relevance feedback approaches. In our test beds, without rank normalization, we would draw the conclusion QPM is slightly better than AND-merge, but OR-merge is much better than the other two, c.f. Fig. 2(a)(b). However, if rank normalization is considered, we would see that the difference is not so large, c.f. Fig. 2(c)-(f). In the CBIR test bed, OR-merge is better than QPM and AND-merge, and the other two are almost the same. While in the TEXT test bed, the three all perform similarly.

The reason why the OR-merge performs better in the CBIR test bed comes from the data distribution of the COREL image collection. In the COREL database, the relevant images (in the same collection) are scattered into several disjoint regions in the perceptual space. For example, for a semantic category flower, there are red, yellow, and white flowers. The OR-merge can form disjoint query regions which fit these areas. But AND-merge and QPM cannot. This makes OR-merge perform better. On the other hand in the text test beds, relevant documents are clustered in some continuous region in space. This makes the three approaches perform quite similarly but OR-merge converges more slowly than the other two approaches initially. In Fig. 2(d)(f), we can see that in the first feedback

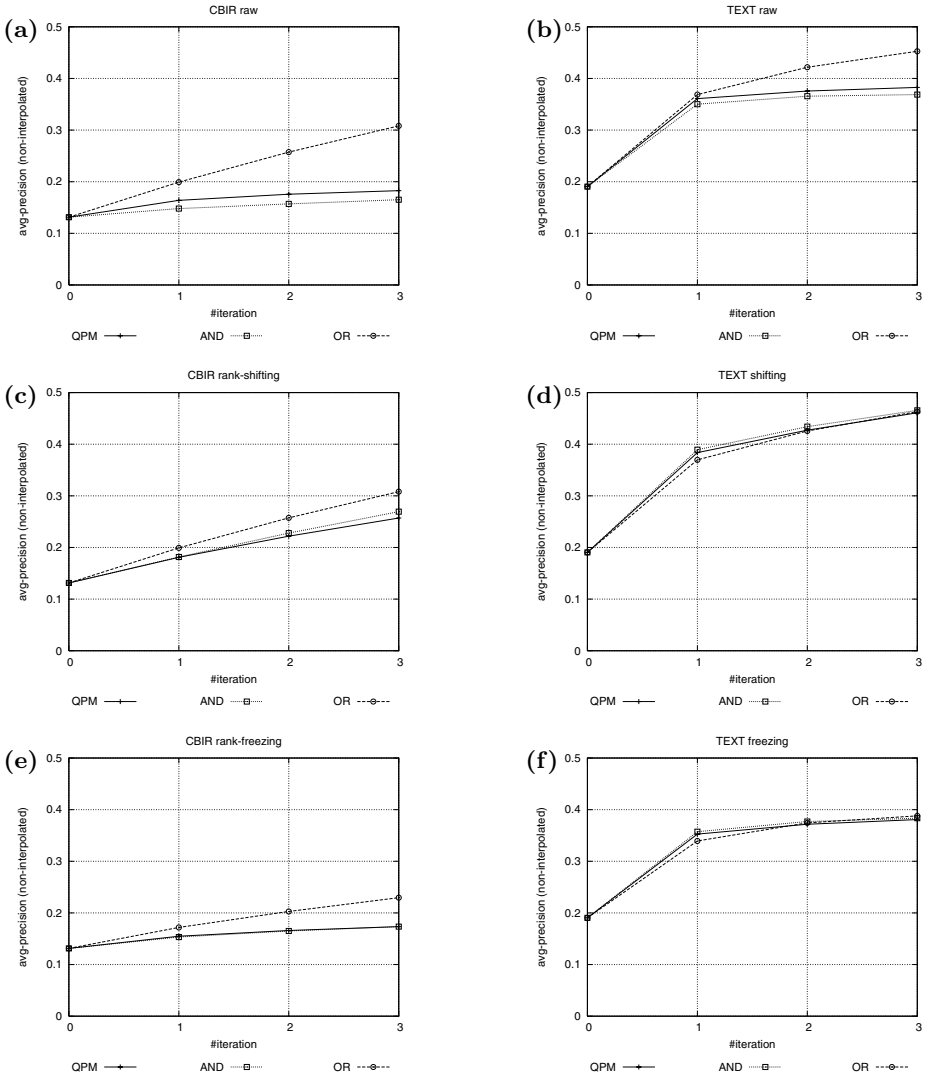


Fig. 2. Comparisons of relevance feedback approaches with/without rank normalization in image and text retrieval environment

iteration, the OR-merge performance line is a little bit lower than the other two. This shows us there is no generally “better” approach, but different approaches are suitable for different applications. For databases where relevant objects are scattered into small clusters, such as a personal digital photo collections, the OR-merge is more suitable to carry out the task. On the other hand, for databases where such clustering phenomenon is not strongly present, like a text collection, QPM is favored because it converges fast and has a light computational cost. If

the retrieval system is a black box or the combine query approach is hard to implement, the AND-merge may be the best approach to choose.

Second, we analyze the performance improvement over iterations. With rank-shifting, looking at QPM and AND-merge, we would draw the conclusion that the performance is greatly improved during feedback iterations. However, with rank-freezing, we would say that in the CBIR test bed, there is almost no improvement in these two techniques after the first feedback iteration. And it is even small in the first iteration. In this case, the improvement mainly comes from those feedback examples which are included in the evaluation.

Consequently, if our purpose is to compare performance among different feedback approaches, it seems that both rank-freezing and rank-shifting can preserve the correct relative order. Since rank-shifting is much easier to be implemented, it is more suitable under this case. But if we want to compare real performance improvement along with feedback iterations, rank-freezing can give us more objective scores.

7 Conclusions

This paper grew out of an effort to understand the performance contribution of several different techniques for incorporating relevance feedback into multimedia information retrieval. Because the techniques came from different domain specialties with very little interaction across communities, the existing work and evaluation was hard to put in context. Consequently, it is difficult to know whether a newly proposed technique does, in fact, make any difference to retrieval effectiveness. We have made an effort here to put the evaluation of relevance feedback techniques in a framework where actual performance can be teased out. We have shown the relationship between multi-query retrieval and relevance feedback and demonstrate a framework in which these systems can be evaluated.

We have also shown very clearly how improper rank normalization can lead to very erroneous conclusions. Rank normalization has been used extensively in text IR but as far as we know, not at all in multimedia evaluation. The incorporation of rank normalization in the retrieval evaluation affords us a better understanding of the retrieval effectiveness of systems employing relevance feedback. This is extremely important if we are to properly understand the behavior of adaptive IR systems.

References

1. Rui, Y., Huang, T., Mehrotra, S.: Relevance feedback techniques in interactive content-based image retrieval. In: *Storage and Retrieval for Image and Video Databases (SPIE 1998)*. (1998) 25–36
2. Porkaew, K., Ortega, M., Mehrotra, S.: Query reformulation for content based multimedia retrieval in mars. In: *Proc. of ICMCS'99, San Diego, CA, USA (1999)* 747–751
3. Wu, L., Faloutsos, C., Sycara, K., Payne, T.: Falcon: Feedback adaptive loop for content-based retrieval. In: *Proc. of VLDB'00, Cairo, Egypt (2000)* 297–306

4. Kim, D., Chung, C.: Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In: Proc. of ACM SIGMOD'03, San Diego, CA, USA (2003) 599–610
5. Ishikawa, Y., Subramanya, R., Faloutsos, C.: MindReader: Querying databases through multiple examples. In: Proc. of VLDB'98. (1998) 218–227
6. Rocchio, J.: Relevance feedback in information retrieval. In Salton, G., ed.: *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall (1971) 313–323
7. Williamson, R.: Does relevance feedback improve document retrieval performance? In: ACM SIGIR'78. (1978) 151–170
8. Liu, W., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M.: Semi-automatic image annotation. In: Proc. of INTERACT'01. (2001) 326–333
9. Jin, X., French, J.: Improving image retrieval effectiveness via multiple queries. In: Proc. of ACM MMDB'03, New Orleans, LA (2003) 86–93
10. Müller, H., Marchand-Maillet, S., Pun, T.: The truth about corel - evaluation in image retrieval. In: CIVR '02. (2002) 38–49
11. Yan, R., Jin, R., Hauptmann, A.: Multimedia search with pseudo-relevance feedback. In: CIVR'03. (2003)
12. Westerveld, T., de Vries, A.P.: Experimental result analysis for a generative probabilistic image retrieval model. In: SIGIR '03. (2003) 135–142
13. Liu, W., Su, Z., Li, S., Y.F.Sun, H.J.Zhang: Performance evaluation protocol for content-based image retrieval algorithms/systems. In: IEEE CVPR Workshop on Empirical Evaluation Methods in Computer Vision. (2001)
14. Fagin, R.: Combining fuzzy information: an overview. In: ACM SIGMOD Record. (2002) 109–118
15. Fox, E., Shaw, J.: Combination of multiple searches. In: Proc. of TREC2. (1994)
16. Belkin, N., Cool, C., Croft, W., Callan, J.: The effect of multiple query representations on information retrieval performance. In: Proc. of ACM SIGIR'03. (1993) 339–346
17. Shaw, J., Fox, E.: Combination of multiple searches. In: Proc. of TREC3. (1995)
18. Salton, G., Fox, E., Wu, H.: Extended boolean information retrieval. In: comm. of the ACM. Volume 26. (1983) 1022–1036
19. Korfhage, R.: *Information Storage and Retrieval*. John Wiley and Sons, New York (1994)
20. Wang, J., Du, Y.: Scalable integrated region-based image retrieval using irm and statistical clustering. In: Proc. of JCDL'01, Roanoke, VA (2001)
21. French, J., Jin, X., Martin, W.: An empirical investigation of the scalability of a multiple viewpoint cbir system. In: Proc. of CIVR'04, Dublin, Ireland (2004) 252–260
22. French, J., Watson, J., Jin, X., Martin, W.: Using multiple image representations to improve the quality of content-based image retrieval. In: Tech. report CS-2003-10, Dept. of Computer Science, Univ. of Virginia. (2003)

An Explorative Study of Interface Support for Image Searching

Jana Urban and Joemon M. Jose

Department of Computing Science, University of Glasgow, Glasgow G12 8RZ, UK
{jana, jj}@dcs.gla.ac.uk

Abstract. In this paper we study interfaces for image retrieval systems. Current image retrieval interfaces are limited to providing query facilities and result presentation. The user can inspect the results and possibly provide feedback on their relevance for the current query. Our approach, in contrast, encourages the user to group and organise their search results and thus provide more fine-grained feedback for the system. It combines the search and management process, which – according to our hypothesis – helps the user to conceptualise their search tasks and to overcome the query formulation problem. An evaluation, involving young design-professionals and different types of information seeking scenarios, shows that the proposed approach succeeds in encouraging the user to conceptualise their tasks and that it leads to increased user satisfaction. However, it could not be shown to increase performance. We identify the problems in the current setup, which when eliminated should lead to more effective searching overall.

1 Introduction and Motivation

Content-based image retrieval (CBIR) systems have still not managed to find favour with the public even after more than a decade of research effort in the field. There are two main reasons for their lack of acceptability: first, the low-level features used to represent images in the system do not reflect the high-level concepts the user has in mind when looking at an image (*semantic gap*); and – partially due to this – the user tends to have major difficulties in formulating and communicating their information need effectively (*query formulation problem*).

The semantic gap is inherent to CBIR [1] and finding better feature representation has been at the core of CBIR research since the early stages. A large variety of features has been proposed over the course of time: from the initial and still widely used low-level features, such as colour and texture, e.g. [2], to more high-level techniques, such as visual templates [3], and finally a combination of visual cues and textual annotations to arrive at “semantic” features, e.g. [4]. Despite this, the current techniques have not succeeded in bridging the semantic gap. A contributing factor is the fact that an image’s meaning is very subjective and context-dependent, which makes it difficult to find generic solutions that do not incorporate the users’ opinions.

The query formulation problem, on the other hand, has emerged as an IR problem in general [5]. The internal representation of documents is optimised for indexing efficiency and retrieval performance, but is, more often than not, rather alien to the user. The semantic gap only amplifies the problems associated with creating a meaningful query that fulfills a user's request.

Hence, improving the way images are represented is only part of the story. In order to assist the user in communicating their requests effectively, better interfaces are needed. The interface should provide a natural means to communicate information needs, should elicit and detect changes in a user's need while interacting with the system, and should in general engage the user in the task they want to solve rather than in the details of how the retrieval system works.

With these requirements in mind, we have proposed a system, EGO, that combines the search and the management process [6]. While searching for images, the creation of groupings of related images is supported, encouraging the user to break the task up into related facets to organise their ideas and concepts. The system can then assist the user by recommending relevant images for selected groups. This way, the user can concentrate on solving specific tasks rather than having to think about how to create a good query in accordance with the retrieval mechanism. It allows the user to interact more directly with the results in a way that is closer to their mental model of solving a search task.

In this paper we present an explorative study comparing two interfaces with respect to the support they offer the user to search for images and organise their results. Our aim is to collect evidence on whether the proposed system helps the user to conceptualise their search tasks. Further, we test our hypothesis that EGO helps to overcome the query formulation problem, since – relying on the in-built recommendation system – there is no need to create a query in order to initiate a search. We measure EGO's success in these two issues compared to a traditional relevance feedback system as a baseline. In the relevance feedback system, the user is given the option of selecting relevant images from the search results in order to improve the results in the next iteration. The evaluation is based on real users, performing practical and relevant tasks and captures a large amount of interaction data, which can be used in follow-up evaluations requiring a long-time involvement of the user.

The remainder of the paper is organised as follows. The interfaces used in the evaluation are described in Section 2. Section 3 sets out the experimental methodology, followed by a detailed analysis of the results and a summarising discussion in Sections 4 and 5. Finally, Section 6 concludes the paper.

2 The Interfaces

As a result of the above requirements, we have designed the EGO system. EGO is a personalised image management and retrieval tool that learns from and adapts to a user by the way they interact with the image collection. The high-level concepts of the EGO system are described in the context of other CBIR systems in [6]. In the experiment we evaluate a simplified version of its interface. A traditional relevance feedback interface serves as baseline.

2.1 Retrieval System

The underlying retrieval system is the same in both interfaces. It involves choosing an ideal query and learning the parameters of the matching function by the user provided examples.

Image Representation. The images are represented according to the hierarchical object model proposed in [7]. In this model an image is represented by a set of feature vectors, one for each distinct feature implemented, rather than a single stacked feature vector.

Implemented Features. We use the following 6 low-level colour, texture and shape features (feature dimension): Average RGB (3), Colour Moments (9) [8]; Co-occurrence (20), Autocorrelation (25), Edge Frequency (25) [9]; Invariant Moments (7) [10].

Distance Measure. The distance between an object x in the database and a given query representation q is computed in two steps. First, the individual feature distances g_i (for i in $1..I$, where I is the number of features) are computed by the generalised Euclidean distance,

$$g_i = (\mathbf{q}_i - \mathbf{x}_i)^T W_i (\mathbf{q}_i - \mathbf{x}_i) \quad (1)$$

where \mathbf{q}_i and \mathbf{x}_i are the i -th feature vectors of the query q and the database object x respectively, and W_i the *feature transformation matrix* used for weighting the feature components. W_i is a $K_i \times K_i$ real symmetric full matrix, where K_i is the i -th feature dimension. The second step is then to combine the individual distances to arrive at a single distance value d . This is achieved by a linear combination between $\mathbf{g} = [g_1, \dots, g_I]^T$ and a feature weight vector \mathbf{u} ,

$$d = \mathbf{u}^T \mathbf{g} \quad (2)$$

The Recommendation System is based on a relevance feedback algorithm, that attempts to learn the best query representation and feature weighting for a selected group of images (positive training samples).

Learning the Feature Weights. We adopt the optimised framework for learning the feature weights proposed in [11]. Due to the hierarchical object model, it distinguishes between intra- and inter-feature weights. The optimal intra-feature component weights are given by an optimal feature space transformation matrix W_i . W_i is calculated as,

$$W_i = \det(C_i)^{\frac{1}{K_i}} C_i^{-1} \quad (3)$$

where C_i is the *weighted covariance matrix* of the N positive examples according to the i -th feature. W_i takes the form of a full matrix, if N is larger than the dimensionality of the feature, otherwise only the diagonal entries are considered. The optimal inter-feature weights $\mathbf{u} = [u_1, \dots, u_I]$ are the weights that best capture the inter-similarity between the training samples. The \mathbf{u}_i 's are solved by,

$$u_i = \sum_{j=1}^I \sqrt{\frac{f_j}{f_i}} \quad (4)$$

where $f_i = \sum_{n=1}^N g_{ni}$. The optimal intra-feature weights W_i and the optimal inter-feature weights u are used in Equations (1) and (2) respectively to calculate the total distance between a database object and the query representation.

Computing the Query Representation and Ranked Results. Our proposed learning scheme relies on a form of query expansion. The chosen query representation for a group is a multi-point query [12], whereby each query point represents one cluster of visually similar images in the group. The query points are selected as the image closest to each cluster centroid, and are weighted relative to the cluster size. When issuing the multi-point query to the system, a separate result list will be returned for each query point, which need to be combined. An investigation of several combination strategies [13] has led us to choosing a rank-based *voting approach (VA)*. Please refer to [13] and [6] for more details.

For the purpose of the evaluation however, we are simply computing one overall query representation as in [11]. This is mainly due to computational complexity (so as not to stretch the users' patience), but also due to some anomalies we found during the evaluation due to the clustering algorithm used.

2.2 Workspace Interface - WS

The combination of retrieval and management system is achieved by providing a workspace in the interface which allows the user to organise their search results. Images can be dragged onto the workspace from any of the other panels (or imported from outside the system) and organised into groups. The grouping of images can be achieved in an interactive fashion with the help of a recommendation system. For a selected group, the system can recommend new images based on their similarity with the images already in the group. The user then has the option of accepting any of the recommended images by dragging them into an existing group.

The interface used in the evaluation is a simplified version of that of the EGO system. EGO has some additional features for personalisation and can, in principle, accommodate any sort of query facility. Since our main objective in these experiments is to evaluate the usefulness of the workspace (and also to avoid biasing the participants by the naming of the experimental systems), this interface is referred to as the Workspace Interface (*WS*). The WS interface depicted in Figure 1 comprises the following components:

1. Given Items Panel: This panel contains a selection of images (three per task) provided for illustration purposes and can be used to bootstrap the search;
2. QBE Panel. This provides a basic query facility to search the database by allowing the user to compose a search request by adding example images to this panel. Clicking on the "Search" button in this panel will issue a search, which causes the system to automatically construct a query from the examples provided and compute the most similar images in the database.
3. Results Panel: The search results from a query constructed in the QBE panel will be displayed in this panel. Any of the returned images can be dragged onto the workspace to start organising the collection or into the QBE panel to change the current query.

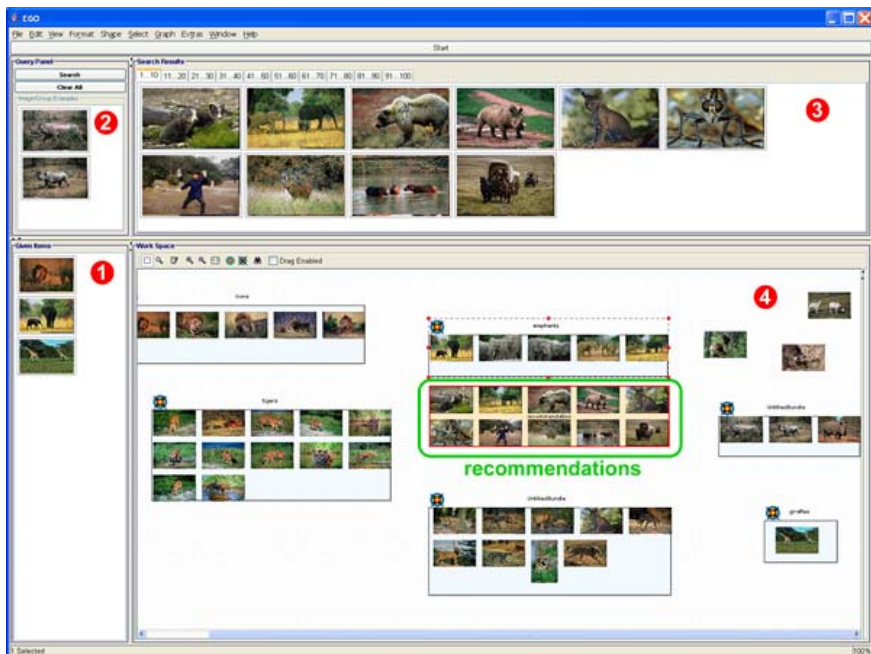


Fig. 1. Annotated WS interface

4. **Workspace Panel:** The workspace holds all the images added to it by the user, and serves as an organisation ground for the user to construct groupings of images. Groupings of images can be created by right-clicking anywhere on the workspace, which opens a context menu in which the option can be selected. Traditional drag-and-drop techniques allow the user to drag images into (or out of) a group or reposition the group on the workspace. An image can belong to multiple groups simultaneously. Panning and zooming techniques are supported to assist navigation in a large information space. Also, the recommendations will be displayed close to the selected group on the workspace (see centre of workspace in Figure 1). So as not to burden the user, the number of recommended images (set to 10 in this evaluation) is based on the standard cognitive limits of 7 ± 2 [14].

To recapitulate, the query facilities available in the WS interface are: (1) manually constructed queries by providing one or more image examples (QBE), and (2) user-requested recommendations.

2.3 Relevance Feedback Interface - CS

The baseline system is a traditional relevance feedback system, referred to as *CS* (for Checkbox System). Relevance feedback (RF) is an automatic process of improving the initial query based on relevance judgements provided by the user [7]. The process is aimed at relieving the user from having to reformulate the

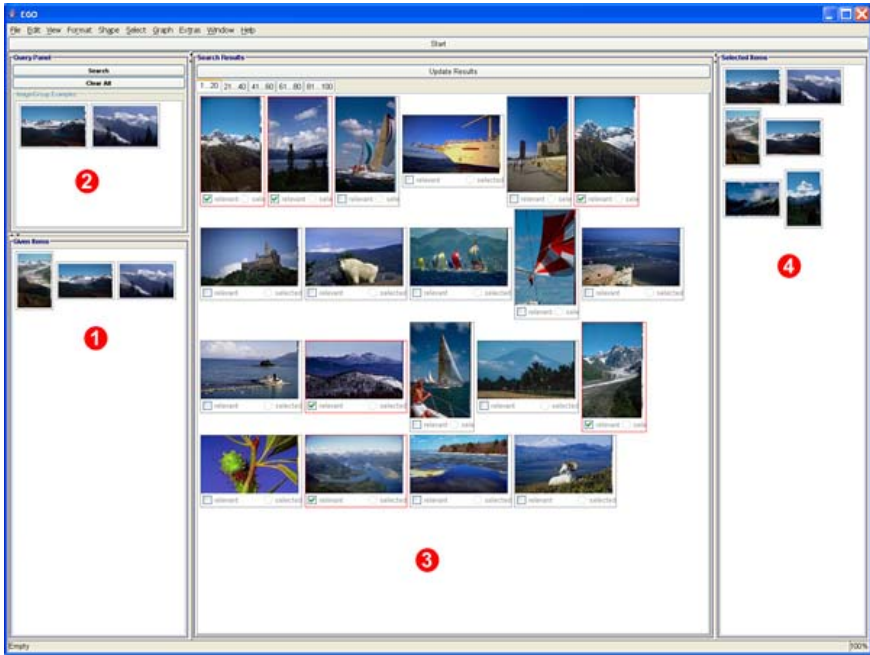


Fig. 2. Annotated CS interface

query in order to improve the retrieval results incrementally. The search becomes more intuitive to the user, since they are only requested to label the returned images as either relevant or not. However, it is still an ongoing research challenge to accurately learn the information need from the user based on a few relevance judgements [15].

Figure 2 shows the CS interface with the following components:

1. Given Items Panel: as above.
2. QBE Panel: as above.
3. Results Panel: As above, but instead of dragging a relevant image onto the workspace the user has the choice of labelling it by selecting a checkbox underneath the image. After relevant images have been marked the user can ask the system to update the current search results (based on the feedback provided) by clicking the “Update Results” button in this panel.
4. Selected Items Panel: Any item selected relevant during the course of the search session will be added to this panel. The user can manually delete images from this panel if they change their mind at a later change.

Finally, CS supports two query facilities: (1) QBE as above, and (2) automatic query reformulation by the user feedback provided in the search results (RF).

3 Experimental Methodology

It has been argued that traditional IR evaluation techniques based on precision-recall measures are not suitable for evaluating adaptive systems [16, 17]. Thus in order to evaluate the systems, we used a task-oriented, user-centred approach [18]. We have designed the experiments to be as close to real-life usage as possible: we have chosen participants with a design-related background and have set tasks that are practical and relevant.

In our evaluative study, we adopted a randomised within-subjects design, in which 12 searchers used two systems. The independent variable was system type; two sets of values of a variety of dependent variables indicative of acceptability or user satisfaction were to be determined through the administration of questionnaires.

To counterbalance the effect of learning from one system to the other, the order of the systems and tasks was rotated according to a Latin square design. For the purpose of the experiment we employed a subset of the Corel collection (CD 1, CD 4, CD 5, and CD 6 of the Corel 1.6M dataset), containing 12800 photographs in total.

3.1 Tasks

In order to place our participants in a real work task scenario, we used a simulated work task situation as conducted in [16]. This scenario allows the users to evolve their information needs in just the same dynamic manner as such needs might be observed to do so in participants' real working lives. A description of the work task scenario and tasks is provided in Figure 3.

Task Scenario

Imagine you are a designer with responsibility for the design of leaflets on various subjects for the Wildlife Conservation (WLC). The leaflets are intended to raise awareness among the general public for endangered species and the preservation of their habitats. These leaflets [...] consisting of a body of text interspersed with up to 4–5 images selected on the basis of their appropriateness to the use to which the leaflets are put.

Category Search Task:

You will be given a leaflet topic from the list overleaf. Your task involves searching for as many images as you are able to find on the given topic, suitable for presentation in the leaflet. In order to perform this task, you have the opportunity to make use of an image retrieval system, the operation of which will be demonstrated to you. You have 10 minutes to attempt this task.

Design Task:

This time, you're asked to select images for a leaflet for WLC presenting the organisation and a selection of their activities (some of WLC's activities are listed overleaf but feel free to consider other topics they might be involved in). Your task is to search for suitable images and then make a pre-selection of 3-5 images for the leaflet. You have 20 minutes to attempt this task.

Fig. 3. Task Description

We created two different tasks: one resembling category search (i.e. users were asked to find as many images as possible from a given topic); and the other resembling an open-ended design task, where they had to search for and make a choice of 3-5 images. The first task was set on both systems, CS and WS, while the latter one was performed on WS only after having completed the category searches. A maximum time was set for all tasks in order to limit the total time spent on the experiment. This was 10 minutes for the category search, and 20 minutes for the design-task.

3.2 Hypotheses

The hypotheses investigated in this study are that the proposed approach for image retrieval and management helps the user to conceptualise their search tasks and to overcome the query formulation problem. The following sub-hypotheses provide more justification:

- Grouping search results on the workspace incites the user to organise results for their search/work task, which in turn helps the user to solve the task. (Organisation as a secondary notation in support of memory/information seeking.)
- The recommendation system helps to overcome the query formulation problem, because it is closer to “real life” search strategies.

In particular we investigate user’s performance on WS compared to a standard relevance feedback interface, CS. The latter relies on relevance assessments provided by the user explicitly by marking images from the search results as relevant. Our assumption is that the relevance assessment in CS might be easier and quicker to use, but is less transparent to the user in comparison to creating groups on the workspace in WS, where the user has control over which images belong together. The option of interactively grouping the search results is assumed to be more natural to the user and to lead to a higher level of control.

3.3 Participants

Since we wanted to test the system in a real-life usage scenario, our sample user population consisted of post-graduate design students and young design professionals. Responses to an entry-search questionnaire indicated that our participants could be assumed to have a good understanding of the search and design task we were to set them, but a more limited knowledge or experience of the search process. We could also safely assume that they had no prior knowledge of the experimental systems.

All participants were in the age group of 20-30 years. There were 9 male participants and 3 female. They had on average 5 years experience in a design-related field (graphic design, architecture, photography). Most people dealt with digital images at least once a day as part of their course or work.

The participants were also asked about their prior experience with search engines and services for searching for images, and image management systems for

organising their own images. Every participant had used an internet image search engine before, whereas only 5 people had used a stock image collection (such as Getty Images, Corbis, Corel). Concerning the organisation of their images, 9 people did not use any management system but just organised their images into folders. The image management systems that were used by the remaining 3 users were ACDSee, Picasa and Extensis Photo Studio.

3.4 Procedure

We met each participant on a separate occasion and adhered to the following procedure:

- an introductory orientation session
- a pre-search questionnaire
- a hand-out of written instructions for the tasks and setting the scenario
- **Part 1:** category search
 - for each system (CS and WS)
 - * a training session on the system
 - * a search session in which the user interacted with the system (max 10 min)
 - * a post-search questionnaire
 - a questionnaire comparing the two systems
- **Part 2:**
 - a search session on WS system (max 20 min)
 - a post-search questionnaire

The total time for one session was 120 min.

4 Results Analysis

There are two objectives of this experiment: (1) to compare the two systems according to their effectiveness and user satisfaction; and (2) to analyse how people make use of the workspace depending on the nature of the tasks. These two parts of the results analysis are expected to shed light on the experimental hypotheses that WS helps users to both conceptualise their tasks better and overcome the difficulties with formulating queries.

4.1 System Comparison

The first objective of the experiment was to compare the two interfaces. It involved two category search tasks, one on each system. The analysis is based on data obtained through questionnaires and usage logs. The questionnaires present a subjective view indicative of the system's acceptability and usability from the users' perspective, while the log data provides a means of judging the task performance objectively. In the questionnaires, we used 5-point semantic differentials, 5-point Likert scales and open-ended questions. Tests (using the non-parametric Wilcoxon Paired-Sample test) for statistical difference will be given where appropriate with $p \leq .05$, unless otherwise stated. The results for the semantic differentials and Likert scales are in the range [1, 5], with 5 representing the best value. \overline{CS} and \overline{WS} denote the means for CS and WS respectively, while \widehat{CS} and \widehat{WS} denote the medians.

Task Performance. Data in the usage logs sheds light on how people actually used the system. From this data we can obtain information on the number of relevant images found over the course of the search session. (The ground-truth was obtained by manually labelling relevant images.) Table 1 shows the number of relevant images for each of the tasks and systems. The total number of relevant images varies greatly per task. The level of recall (number of relevant images found over number of total relevant images for the task) attained depends therefore not only on the complexity of the task but also on the number of relevant images available in the system. The tasks were chosen so that Tasks 1-3 represented simple and concrete topics (“mountains”, “tigers”, “elephants”), while Tasks 4-6 comprised multiple facets (“animals in the snow”, “African wildlife”, “underwater world”). Looking at the data in Table 1 it can be inferred that users generally performed better in CS independent of the nature of the task.

Table 1. Number of relevant images found and corresponding levels of recall per task

System	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	All
Total #Relevant Images	549	114	103	220	865	402	375.5
#Images found AVG	56.5	14.0	15.25	44.0	38.75	36.75	34.2
#Images found CS	71.5	18.0	18.5	54.5	50.5	34.0	41.2
#Images found WS	41.5	10.0	12.0	33.5	27.0	29.0	25.5
Recall AVG	10.3%	12.3%	14.8%	20.0%	4.5%	7.8%	11.6%
Recall CS	13.0%	15.8%	18.0%	24.8%	5.8%	8.5%	14.3%
Recall WS	7.6%	8.8%	11.7%	15.2%	3.1%	7.2%	8.9%

User Satisfaction. After having completed a task the participants were given a questionnaire about their search experience (post-search questionnaire). Finally, they were asked to compare the two systems in the exit questionnaire. In this section we will analyse the users’ opinion on the systems as inferred from the answers provided in the questionnaires.

Post-Search Questionnaire. In the post-search questionnaire people were asked about the task they performed, the images received through the searches, and the system itself.

Task and Search Process. In general, the tasks were considered *clear* and *familiar*, but slightly more *simple* in CS (see Table 2). The search process was considered slightly more *relaxing* and *easier* in CS, but significantly more *interesting* in WS. However, people tended to agree more with the statement that they had enough time to complete their task in CS: $\overline{CS} = 4.6$, $\overline{CS} = 5$ and $\overline{WS} = 4.3$, $\overline{WS} = 4$.

Images. The images received through the searches were considered equally *relevant* and *appropriate*, but significantly more *complete* in WS (see Table 4). More people agreed with the statement, that they discovered more aspects of the category than initially anticipated during the search on WS ($\overline{CS} = 2.4$, $\overline{CS} = 2$ and $\overline{WS} = 4.4$, $\overline{WS} = 5$; $p = 0.02$). On the other hand, people tended to be equally satisfied with their search results in both systems ($\overline{CS} =$

Table 2. Semantic Differentials Results for the Task and Search Process Part

Differential	\overline{CS}	\widetilde{CS}	\overline{WS}	\widetilde{WS}	p
clear	4.8	5	4.8	5	-
familiar	3.8	4	3.7	4	-
simple	4.8	5	4.5	5	-
relaxing	4.6	5	3.9	4	-
easy	4.5	5	4.3	5	-
interesting	3.6	4	4.3	4	0.016

Table 3. Semantic Differentials Results for the System Part

Differential	\overline{CS}	\widetilde{CS}	\overline{WS}	\widetilde{WS}	p
wonderful	3.7	4	4.1	4	-
satisfying	3.9	4	4.1	4	-
stimulating	3.2	3	3.8	4	0.004
easy	4.6	5	4.1	4	0.031
flexible	2.8	3	3.9	4	0.001
novel	3.1	3	4.2	4	0.016
effective	4.3	4	4.3	4	-

Table 4. Semantic Differential Results for the Images Part

Differential	\overline{CS}	\widetilde{CS}	\overline{WS}	\widetilde{WS}	p
relevant	4.2	4	4.2	4	-
appropriate	4.2	4	4.3	4	-
complete	3.3	3	4.1	4	0.027

Table 5. Likert Scale Results for System Part

Statement	\overline{CS}	\widetilde{CS}	\overline{WS}	\widetilde{WS}	p
learn to use	4.8	5	4.1	4	0.03
use	4.5	5	4.0	4	-
explore collection	3.3	3	4.3	4	0.03
analyse task	3.1	5	4.5	5	0.02

Table 6. Comparison of system rankings

System	(a) learn	(b) use	(c) effective	(d) liked best
CS	5	5	4	3
WS	3	6	6	8
no difference	4	1	2	1

3.6, $\widetilde{CS} = 4$ and $\overline{WS} = 3.6$, $\widetilde{WS} = 4$). There is no apparent correlation between actual task performance and perceived task performance.

System. The users considered CS significantly more *easy* than WS, while they considered WS to be significantly more *stimulating*, *flexible*, and *novel*. Table 3 shows the results for these differentials.

People found CS significantly easier to *learn to use*, while there was only a marginal difference between *using* them. However, people thought WS helped them to explore the collection better, as well as analyse the task better. The results for the responses to these statements are provided in Table 5.

Exit Questionnaire. After having completed both category search tasks having used both systems, the users were asked to determine the system that was (a) easiest to learn to use, (b) easiest to use, (c) most effective, and (d) they liked best overall. Table 6 shows the users’ preferences of systems for each of the statements. It shows that, while it is easier to learn to use CS, the majority of people preferred WS and found it more effective.

In open-ended questions, the participants were invited to give their opinion on what they liked or disliked about each system. The advantages listed

for CS were that it was fast, efficient and easy to use. Its disadvantages included that the users felt they did not have enough control over the search and that its interface was less intuitive. In WS, people liked the ability to plan their searches by organising the results into groups, and the overview they had of the results and searches that the organisation brought along. In addition, the system's flexibility and more control options were noted as advantages. The disadvantages were mainly concerned with the poor quality of the recommendations and that the handling of groups was sometimes cumbersome. Both of these issues are not inherent in the interaction paradigm of the proposed system itself, and can consequently be improved or even avoided in the future. The recommendation quality can be improved by a better choice of visual features and also by recommendations based on other people's groupings. The handling of the groups and images within groups is a matter of programming.

4.2 Task Analysis

The second objective of the study is to judge the usefulness of the workspace to help the user to conceptualise their task. In order to find out how people make use of the groupings and organise their workspace, we have created two different task scenarios in the experiment: the category search scenario and the design task scenario. The former (set on both WS and CS) aims at maximising recall, while the latter aims at finding a selection of good quality images that work well together (only on WS). By analysing the number of groups created and the average number of images per group for the various tasks, we can identify how these numbers relate to task complexity.

Unfortunately, we cannot present the full analysis in this paper. Please refer to [19]. To summarise, we found a correlation between the number of groups created and the complexity of the task set. Further, responses in the questionnaires showed that the management of search results was deemed more helpful in the design scenario, which is more flexible and open to interpretation than the category search scenario. In the category search scenario, the usefulness of the organisation also depended on the complexity of the task: the more facets the task comprised, the more useful the workspace was considered. This strong dependency between both the number of groups created and the users' perception of the workspace's usefulness, led us to the conclusion that our approach indeed helps in conceptualising the task.

5 Discussion

By analysing users's behaviour in different task scenarios, we have been able to show that the grouping facility was used to reflect the various task facets, and therefore helped to conceptualise tasks. On the other hand, it is more difficult to draw a definite conclusion on the second hypothesis, namely that our approach helps to overcome the query formulation problem. The responses in the questionnaires suggest that the search process is more interesting in WS, the system

helped them to discover more aspects of the task, and found it more stimulating, flexible and novel. In general, they preferred WS over CS and found it more effective for the task. The participants particularly liked the ability to plan their searches and organise their results. In comparison, they considered they were lacking control over their searches in CS. However, the actual task performance does not reflect the users' perception. The number of relevant images found per task were generally higher in CS than in WS. Based on the analysis of the questionnaire data above, the reason for this is that the selection of relevant images is much faster than the dragging of images. Also, the users spent time on creating groups of images and moving images between groups in the WS system. Since we have set a maximum time limit, the number of images found was generally higher in CS, where the user was not "distracted" by managing their search results.

In addition, the failure of the recommendation system has most probably contributed to these results. Analysing the users' comments, we could identify that many people thought the recommendation system would potentially have been a very useful feature, but was not employed due to its inability to recommend relevant images. Our initial hypothesis, namely that the recommendation system helped to overcome the query formulation problem, could not be verified directly. On the other hand, when analysing the way the users manually created the queries, we could observe an interesting pattern. They usually started off with a small number of example images (from the given items, and some initial results). Once they had created a group on the workspace that contained a number of relevant images, they used the whole group in the QBE search to find similar images to the *group*. We assume that, had the recommendation system worked better, users would have used the recommendations in that case. However, since this was not the case, they had to resort to the manual facility of finding more similar images for the group.

In conclusion, the difference in performance can be attributed to the additional effort – both physical (slower selection process) and cognitive – required in WS. While the users commented on the additional physical effort, they did not perceive the additional cognitive effort as negative. On the contrary, they thought the organisation to be supportive for solving their tasks as well as potentially beneficial for others to use in the future.

6 Conclusion and Future Work

We have presented a user study comparing the proposed EGO system to an image retrieval system with relevance feedback capabilities. While the performance in a category search task was generally higher in the relevance feedback system, the proposed system led to a higher user satisfaction. We identified possible reasons for the differences in task performance: the time restriction was limiting and the recommendation system's performance was not good enough. Still, the participants preferred our system, because it allowed them to organise their search results and hence conceptualise the task better.

Since we have encountered differences in the perceived usefulness of the grouping facility depending on the task nature, we believe the interface should have a way to be tailored to these contrasting requirements to adapt to its users. In the future, the user should be assisted in determining task aspects and create groups (semi-) automatically. For a multi-aspect task, we could then group results into the various aspects and present recommendations for each group.

Moreover, a more sophisticated active learning approach, such as the one proposed in [20], could help to improve the recommendations based on the visual features of the images. In addition, the recommendations should also incorporate information from the groups created by the users. This can be used to learn associations between images, and, when combined with the visual similarity, lead to not only more accurate recommendations but also to personalised recommendations. This is the case, since similarity between images would then be based on semantic concepts as defined by the users. We would also like to investigate EGO in a collaborative context. By placing the resulting groups of images on a workspace, the user creates traces of their activities. These traces could be used in a collaborative environment in two ways: first, the system can use the groups created by various users to learn general and personal associations between images; and second, by inspecting someone else's workspace one can retrace their activities.

The real benefits of such a management system will only have an effect if it is used over a longer period of time. The organisation of the collection created over time is an important clue for the system to learn and improve its recommendations over time. The interaction data collected in this study will therefore be useful in follow-up evaluations requiring a long-time involvement of the user.

References

1. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence* **22** (2000) 1349–1380
2. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. *Computer* **28** (1995) 23–32
3. Lim, J.H.: Learnable visual keywords for image classification. In: *Proc. of the ACM Int. Conf. on Digital Libraries (DL-99)*, ACM Press (1999) 139–145
4. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: *Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR03)*. (2003) 119–126
5. ter Hofstede, A.H.M., Proper, H.A., van der Weide, T.P.: Query formulation as an information retrieval problem. *The Computer Journal* **39** (1996) 255–274
6. Urban, J., Jose, J.M.: EGO: A personalised multimedia management and retrieval tool. *Int. Journal of Intelligent Systems (IJIS)*, Special Issue on 'Intelligent Multimedia Retrieval' (2005) to appear.
7. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **8** (1998) 644–655

8. Stricker, M., Orengo, M.: Similarity of color images. In: Proc. of the SPIE: Storage and Retrieval for Image and Video Databases. Volume 2420. (1995) 381–392
9. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. 2nd edn. Brooks and Cole Publishing (1998)
10. Hu, M.K.: Visual pattern recognition by moment invariants. *IEEE Trans. Information Theory* **8** (1962) 179–187
11. Rui, Y., Huang, T.S.: Optimizing learning in image retrieval. In: *IEEE Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR-00)* (2000) 236–245
12. Porkaew, K., Chakrabarti, K., Mehrotra, S.: Query refinement for multimedia similarity retrieval in MARS. In: *Proc. of the ACM Int. Conf. on Multimedia* (1999) 235–238
13. Urban, J., Jose, J.M.: Evidence combination for multi-point query learning in content-based image retrieval. In: *Proc. of the IEEE 6th Int. Symposium on Multimedia Software Engineering (ISMSE'04)* (2004) 583–586
14. Miller, G.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review* **63** (1956) 81–97
15. Zhou, X.S., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. *ACM Multimedia Systems Journal* **8** (2003) 536–544
16. Jose, J.M., Furner, J., Harper, D.J.: Spatial querying for image retrieval: A user-oriented evaluation. In: *Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'98)*, ACM Press (1998) 232–240
17. de Vries, A.P.: The role of evaluation in the development of content-based retrieval techniques. Technical Report TR-CTIT-00-19, Centre for Telematics and Information Technology (2000)
18. Ingwersen, P.: *Information Retrieval Interaction*. Taylor Graham, London (1992)
19. Urban, J., Jose, J.M.: Exploring results organisation for image searching. In: *Proc. of INTERACT 2005*, Springer (2005) to appear.
20. Jin, R., Chai, J.Y., Si, L.: Effectiv automatic image annotation via a coherent language model and active learning. In: *Proc. of the ACM Int. Conf. on Multimedia*, ACM Press (2004) 892–899

Context-Based Image Similarity Queries*

Ilaria Bartolini

DEIS - IEIIT-BO/CNR, University of Bologna, Italy
ibartolini@deis.unibo.it

Abstract. In this paper an effective context-based approach for interactive similarity queries is presented. By exploiting the notion of image “context”, it is possible to associate different meanings to the same query image. This is indeed necessary to model complex query concepts that, due to their nature, cannot be effectively represented without contextualize the target image. The context model is simple yet effective and consists of a set of significant images (possibly not relevant to the query) that describe the semantic meaning the user is interested in. When feedback is present, the query context assumes a dynamic nature, changing over time depending on the actual retrieved images judged as relevant by the user for her current search task. Moreover, the proposed approach is able to complement the role of relevance feedback by persistently maintaining the query parameters determined through user interaction over time and ensuring search efficiency. Experimental results on a database of about 10,000 images show the high quality contribution of the proposed approach.

1 Introduction

Advances in the computer technologies and the advent of the Word-Wide Web have produced the explosion of an increasing number of complex data such as digital images, video, and audio. As a primary consequence, there is a pressing need for the definition of efficient and effective techniques able to retrieve such information based on their content.

The traditional paradigm for the retrieving of images is based on keyword annotation. In this approach, human experts manually annotate each image with a textual description, so that text-based information retrieval techniques can be applied [6]. This approach has the advantage of inheriting efficient technologies developed for text retrieval, but is clearly impracticable for the case of very large image DBs. Moreover, its effectiveness highly depends on the subjective opinions of the annotators, who are also likely to supply different descriptions for the same image.

To overcome above difficulties, in the early 1990’s an alternative approach has been proposed. Content-Base Image Retrieval (CBIR) uses visual properties (*features*) to represent the image content. This approach has a wider applicability, since features can be computed automatically, and the information used during the retrieval process is always consistent, since it does not depend on human interpretation. To characterize each image, CBIR systems define a set of low level relevant features able to effectively characterize the content of the images and then use such features for retrieval purposes [8]. The features should be “simple enough” to allow the design of automatic

* This work is partially supported by the WISDOM MIUR Project.

extraction algorithms, yet “meaningful enough” to capture the image content. Under this view, each image is typically represented by a high-dimensional *feature vector*, whose dimensionality depends on the number and on the type of extracted features, and similarity between images is assessed by defining a suitable distance function on the resulting feature space.

CBIR systems, however, assume that high level concepts (as perceived by humans) can be perfectly mapped to low level features (as extracted by the computer): This, of course, may be not always true. This mismatch between human-perceived and computer-provided image representations is known as the “semantic gap” problem [8] and is one of the most challenging problem for multimedia information retrieval.

Although approaches based on a-priori classification of images [9] and on analysis of (possibly available) surrounding text/captions [10, 14] might help in alleviating the semantic gap, they are not always applicable for heterogenous image collections. Moreover, such approaches are not able to “contextualize” the search based on current user needs. Indeed, even a same image might represent different meanings to different users (or to a same user at different times). For example (see [7]), a portrait can suggest the notion of “painting”, when placed in the context of other painting images, and the meaning of “face”, when the context becomes a set of people photos.

Motivated by above observations, in this paper we investigate the potentialities of an approach to contextualize image queries with the aim to solve, or at least alleviate, the semantic gap problem. The key idea is to complement the image query with a set of (possibly not relevant) images able to direct the search to the correct semantic concept (see Section 3 for a real example). Even if our approach is very simple, it is indeed effective and does not require neither a-priori classification, nor textual information associated to images. Furthermore, it can easily complement available feedback techniques [3] by providing a better starting point for the search of complex semantic concepts and a beneficial inertial behavior on the updating of query parameters in the user-system interaction process. This is made possible by exploiting the history of the “best” relevant examples over time. Finally, to ensure search efficiency, our solution can be easily integrated with techniques for learning user preferences (e.g., [1, 13, 15]), by maintaining optimal parameters for each user query. In this way, the main limit of traditional relevance feedback techniques, consisting in “forgetting” user preferences across multiple query sessions, is no more a concern.

Our experiments, conducted on a dataset of about 10,000 images, show that the quality of results obtained from our approach, as measured in term of classical *precision*, outperforms modern interactive retrieval techniques. As for the efficiency, we integrate our context-based method in **FeedbackBypass** [1], and experimentally prove how the number of search interactions needed to reach a given level of precision is reduced.

The rest of the paper is organized as follows. Section 2 surveys some approaches to context-based image similarity. In Section 3 we describe our approach by defining the notion of context. The case when user feedback is present is contemplated in Sections 4 and 5, where an accurate description of the context updating is also provided. In Section 6 we present experimental results showing the effectiveness and the efficiency of our approach. Finally, Section 7 concludes the paper and suggests directions for future work.

2 Context-Based Queries

To the best of our knowledge, no other work has attempted to analyze the effect of using the notion of context, as a set of possibly not relevant images evolving over time, for content-based similarity queries. However, many works share our main goal (i.e., to alleviate, if not completely solve, the semantic gap problem) even if they usually associate a different meaning to the word “context”. In this section, thus, we only survey the contributions of such works, classifying them into three main classes:

Analysis of surrounding text. Here the context is defined as the description of the image content that comes from sources other than its visual properties. Typically such content is expressed in term of *textual* information (e.g., [10, 14]) that comes from manually annotations (e.g., keywords, descriptions, etc.), or surrounding text that is available with the image (e.g., captions, nearby text from Web pages containing the image, subtitles, etc.). The similarity between images is then assessed by also taking into account similarity between associated texts, using standard text retrieval techniques [6]. In details, in [14] the authors propose an image retrieval system that combines visual (i.e., content) and textual (i.e., context) querying at a semantic level, finding a semantic association between low level features and high level concepts. Authors in [10] do the same by also integrating in the search process the notion of *form* of a multimedia document defined as the internal structure of a document (e.g., objects of an image, frames in a video, and chapters of a book).

However, in general a textual information might not be available for every image, or it might not be meaningful to correctly describe the image. This represents the main limit of this context definition.

Taxonomy-based search. The context is represented by means of subject classes (i.e., an ontological concept that represents the semantic content of an image) and by the corresponding definition of a *taxonomy* that arranges such classes into a is-a hierarchy. In this scenario, the search process starts by first browsing the taxonomy, then a classical content-based retrieval is applied. In particular, the WebSeek system [9] provides a powerful semi-automatic approach for classifying images and videos on the Web according to a general subject taxonomy based on text associated with those images and videos, as, for example, Web addresses (or URLs), HTML tags, and hyperlinks between them. Thus, a user looking for images of comets has to browse the “astronomy” category in the taxonomy, and eventually reach the “astronomy/comets” class.

This approach suffers the same limitations above described, in that a classification may not always be available. Furthermore, due to different meanings that a same image/video can assume depending on the particular user and context in which it is collocated, it would be natural to assign it to different semantic classes. The main consequence is that if the selected class in the first phase is not correct, the result of the final search is not accurate.

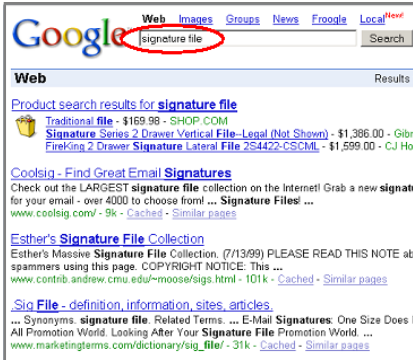
Analysis of objects within a same image. This is the notion of context usually assumed in research areas of image retrieval, such as computer vision. In this case, images are characterized by means of local pictorial objects and their spatial relationships. This is due to the fact that such relationships are able to capture the most relevant part of semantic information in an image. Relation-based representations require that objects

are represented by symbols [11]. Each object is replaced by a symbolic label located at a set of object representative points. In this way, both spatial and topological relations are supported. Relevant examples are: Object A is on object B, A is on the right/left side of B, A disjoints B, A contains B, etc.. This usually requires the use of object recognition techniques that severely limit the applicability of this approach to the case of large heterogeneous image collections, i.e., the only ones we take into consideration.

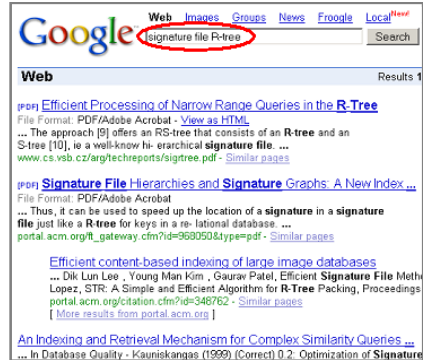
The El Niño image browsing interface [7], even if it does not represent an image retrieval system, is also relevant to our work. When browsing with El Niño, the user implicitly defines semantic concepts by selecting relevant images from a set of randomly displayed objects, and by placing them on the screen so as to reflect their (intended) mutual similarities. This is similar to the definition of our context, since the system adapts its (internal) similarity criterion by using the placement of images, so that the updated display of images matches the similarity intended by the user.

3 Our Approach

The basic idea of the proposed approach is to contextualize user queries by associating a set of objects (defining a *context*) that are possibly not relevant to the queries but that can be helpful to the system in solving, or at least alleviating, the semantic gap problem.



(a)



(b)

Fig. 1. Google results for the general concept “signature file” (a) and the contextualized concept “signature file R-tree” (b)

To give an intuitive example, let us consider a typical Web search scenario where a user is looking for documents related to “access methods” of type “signature file”. If the user adopts the well known search engine *Google*¹ and enters the keywords “signature file”, she obtains the results shown in Figure 1 (a). As it can be observed, none of the

¹ Google: <http://www.google.com>.

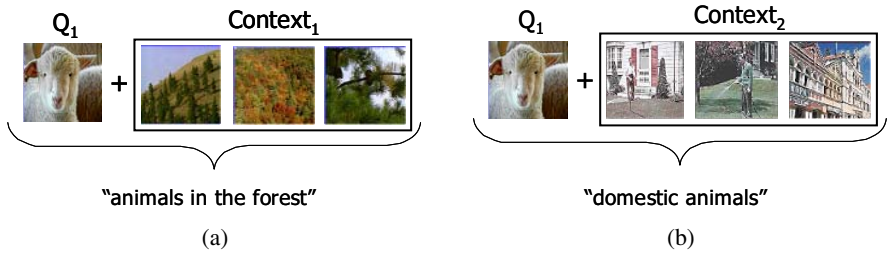


Fig. 2. The same query associated to a context that suggests (a) “animals in the forest” and (b) “domestic animals”

retrieved documents is relevant with respect to the query.² This is due to the different meanings associated to the string “signature file”, i.e., a file containing a signature or an access method. Thus, the problem is: How to tell the system that we are looking for access methods and not for file of signatures? A simple yet effective solution is to add some additional terms, e.g., “R-tree”, that are not directly relevant to the query, but that helps to define a context, i.e., access methods (see Figure 1 (b) for the contextualized result).

In the image domain, the idea is to start from an image query complemented by some other (possibly not relevant) images that contextualize the query by associating the right semantic concept. Even if the approach is simple, it is indeed an effective query model, more flexible with respect to the usual Query By Example (QBE) paradigm. In this scenario, in fact, the same query can be used to retrieve different semantic concept images. This is particular important when user preferences are exploited during the search (i.e., when applying relevance feedback techniques). A simple example is shown in Figures 2, where the same “sheep” image query is used to formulate two completely different searches. This is possible by defining two contexts (in the example, these are represented by means of three images): $Context_1$ (Figures 2 (a)) represents the concept “forest”, whereas $Context_2$ (Figures 2 (b)) the concept “domestic images”. It is possible to observe how, in this particular examples, none of the context images are relevant to the query concepts (i.e., none of them represent animals). However, context images allow a better definition of the “query parameters” used in the search process (as confirmed by experimental results in Section 6).

More precisely, we frame our discussion in the context of the *vector space* model, where an image is represented by a point \mathbf{p} in a D -dimensional space of features, $\mathbf{p} = (p[1], \dots, p[D])$. Given two points, \mathbf{p} and \mathbf{q} , their similarity is measured by means of a distance function d on such space. In details, we adopt the weighted Euclidean distance ($L_{2\mathbf{W}}$, with $w[i] \in \mathbf{W}$ defaults to 1, $\forall i \in [1, D]$).

The usual approach for a user is to submit a query $Q = (\mathbf{q}, k)$, where \mathbf{q} is the query point and k represents the number of results to be returned by the system. Using a default distance function d , \mathbf{q} is compared with the database objects and the k objects which are closest to \mathbf{q} according to d are returned (i.e., $Result(Q, d) = \{\mathbf{p}_1, \dots, \mathbf{p}_k\}$). Thus, the query point \mathbf{q} and the default weights determine the final results.

² For lack of space, we report in Figure 1 the first four documents of the complete result; however, the same behavior is also confirmed by the other, not shown, documents.

However, as also shown in the examples of Figure 2, when the query concept becomes more complex, a single object (i.e., the query point \mathbf{q}) is not able to represent it well. To overcome such limitation, we propose a new query model where the notion of *context* plays the main role. In the new scenario, a query Q is defined as a triple $Q = (\mathbf{q}, \mathcal{C}, k)$, where $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$ is a set of points in the feature space (possibly not relevant to the query) that contextualize \mathbf{q} . The problem faced by the context-based approach can be precisely formulate as follows:

Problem 1. *Given a query point \mathbf{q} and a context \mathcal{C} , determine the distance function $d^{\mathcal{C}} (\equiv \mathbf{W}^{\mathcal{C}})$ able to reflect the intrinsic meaning of images in \mathcal{C} . The equivalence highlights that $d^{\mathcal{C}}$ is the distance function obtained when the weights are set to $\mathbf{W}^{\mathcal{C}}$.*

The problem, thus, consists in reshaping the distance function in order to obtain more accurate results wrt those retrieved using the default distance d . The rationale is that objects in \mathcal{C} are possibly not relevant; thus, they are excluded in the definition of the query point, that remains unaltered, but they are involved in the distance function re-weighting process.

4 Context Evolution

In the previous section we have focused on the first round of the search process. We now consider the possibility, for the user, to provide an evaluation of the relevance of the result objects. In particular, we suppose that, after receiving the result for her query, the user provides her positive/negative feedback on the objects in $Result(Q, d)$. Then, a new query $Q_{new} = (\mathbf{q}_{new}, k)$ and a new distance function $d_{new} (\equiv \mathbf{W}_{new})$ are computed to determine the second round of results. In this way, a feedback loop process takes place until the user is satisfied with the final result. In such context, the results of each round of search only depend on \mathbf{q}_{new} and \mathbf{W}_{new} , that we refer to as *query parameters*.

In this scenario, we rely upon two basic strategies for learning from relevance feedback: the *query point movement* (QPM) and the *re-weighting* techniques. The former concerns the computation of an ideal query point by moving towards the positive results and away from the negative examples. A well-known implementation of this idea has been proposed by Rocchio [6] in the context of information retrieval. Given two sets \mathcal{R} and \mathcal{N} of relevant and not relevant objects, the new query point is adapted as:

$$\mathbf{q}_{new} = \alpha \mathbf{q} + \beta \left(\frac{1}{|\mathcal{R}|} \sum_{\mathbf{p} \in \mathcal{R}} \mathbf{p} \right) - \gamma \left(\frac{1}{|\mathcal{N}|} \sum_{\mathbf{r} \in \mathcal{N}} \mathbf{r} \right) \quad (1)$$

where α , β and γ are weighting factors, satisfying the constraint $\alpha + \beta + \gamma = 1$. More recently, QPM has been applied in several image retrieval systems [5, 3].

The re-weighting strategy updates the distance function by enhancing the feature components, which are more important than others in determining whether a result point is relevant or not. In an early version of the MARS system [5], the authors propose a re-weighting approach based on the standard deviation of the positive examples (i.e., $w[i] = 1/\sigma[i]$). Later on, it was proven in [3] that the optimal choice of weights is to have $w[i] \approx 1/\sigma[i]^2$. The intuition behind using standard deviation and variance is

that a large variance among the values of positive objects in a dimension means that the dimension poorly capture the user information need; viceversa, a small variation indicates that the dimension is important to the user expectation and should carry a higher weight.

When user feedback is present, it is reasonable to consider that the query context \mathcal{C} might be refined/changed depending on the actual retrieved and relevant images found so far. The rationale is that the set of objects defining \mathcal{C} have the primary role to promote an effective discover of a first set of results, i.e., to increase the number of relevant objects in the first round of search. This is indeed an important issue as it represents a better starting point for the application of relevance feedback techniques wrt the results obtained by means of a default distance. However, in our approach the context has a further important role that consists in maintaining the “inertia” of the weights during the re-weighting process, inspired by the inertial behavior of the “Rocchio-like” modification of the query point. Thus, the context has a dynamic nature and has to be properly updated over time. The main idea is to update \mathcal{C} at each round of search (starting from the first round of results) by promoting the exclusion from \mathcal{C} of l selected objects and the inclusion in \mathcal{C} of l specific relevant objects, as dictated by policies that we will describe in the following.

With this in mind and by supposing to use only positive feedback, Problem 1 can be re-formulated as follows:

Problem 2. *Given a query point \mathbf{q} , a current context \mathcal{C} and a current set of relevant objects \mathcal{R} , determine the optimal query parameters $(\mathbf{q}_{\text{new}}, \mathbf{W}_{\text{new}}^{\mathcal{C}, \mathcal{R}})$ for \mathbf{q} , where \mathbf{q}_{new} is the new query point based on \mathcal{R} , and $\mathbf{W}_{\text{new}}^{\mathcal{C}, \mathcal{R}}$ are the new weights computed by means of objects in \mathcal{C} and \mathcal{R} .*

As for the computation of the new query point, we follow the usual QPM approach (see Equation 1). Before entering into details of the re-weighting process, we first describe how the context evolves over time.

4.1 Context Switch

The context switch technique allows to update \mathcal{C} depending on the relevant images in the search result. To this end, distances between context images and relevant images and between all the relevant examples are computed. Depending on the particular “selecting” policy, it is possible to assert which are the images that have to leave the context and the relevant examples that have to replace them. In particular, we provide two different policies:³

Near-selection. The l context images that are farthest (i.e., whose average distance is higher) to all the positive examples have to leave the context and are replaced by the l relevant examples which are closer to all the other relevant examples. The rationale is simple yet effective: Images kept in \mathcal{C} over time are those which are “close” to the positive examples and, thus, have a high probability to be relevant

³ The two strategies are inspired by the heuristic techniques of *near* and *distant expansion* proposed in [4] aiming to change the set of query points in the context of the multi-point query expansion relevance feedback approach.

for the search. This strategy implies that after a certain number of rounds (that is proportional to l and the cardinality of \mathcal{C}) the context contains only relevant images that represent the main semantic concept at a high level of granularity.

Distant-selection. Conversely, the l context images that are closest (i.e., whose average distance is smaller) to all the positive examples have to leave the context and are replaced by the l relevant examples which are farthest to all the other relevant examples. The rationale here is that context images that are close to positive examples probably contain similar information; thus, they are considered redundant and are discarded. On the other hand, images that are distant to the positive examples are kept over time in \mathcal{C} , because they are considered more discriminant. As a consequence, there is no guarantee for images in \mathcal{C} to become all relevant with respect to the semantic concept.

4.2 Context Re-weighting

Due to the dynamic nature of the context, and in order to better represent the information represented by each image in \mathcal{C} , we introduce a *local* context weight w_{c_j} for each image c_j that defaults to $1/m$ (m is the number of context images) and that is updated at each search round. In particular, we define two strategies for its re-computation:⁴

Maximum distance. The lowest weight is associated to the image c_j that is more distant to the positive examples. In details, for each c_j the maximum distance among the positive results is computed as follows:

$$\Delta c_j = \max_{\mathbf{p} \in \mathcal{R}} \{d(c_j, \mathbf{p})\} \quad (2)$$

Minimum distance. Conversely, the highest weight is associated to the image c_j that is closest to the positive examples. For each c_j the minimum distance among the positive results is computed as follows:

$$\Delta c_j = \min_{\mathbf{p} \in \mathcal{R}} \{d(c_j, \mathbf{p})\} \quad (3)$$

Then, the weight w_{c_j} is computed as:

$$w_{c_j} = \frac{\sum_{h \neq j} \Delta c_h}{\sum_h \Delta c_h} \quad (4)$$

In this way, if Δc_j^* is the maximum/minimum value of Δc_j , we are guaranteed that its corresponding weight $w_{c_j^*}$ is the lowest/highest, respectively.

Moreover, a *global* context weight $w^{\mathcal{C}}$ is computed by considering the whole set of context images. In details, the i -th weight component is:

$$w^{\mathcal{C}}[i] = \frac{|\mathcal{C}|}{\sum_{c_j \in \mathcal{C}} w_{c_j} (c_j[i] - \bar{c}[i])^2} \quad (5)$$

where $\bar{c} = \sum_{c_j \in \mathcal{C}} c_j / |\mathcal{C}|$ is the average vector of context images.

⁴ The two solutions take inspiration from the *Maximum distance strategy* described in [4] aiming to solve the query re-weighting problem for multiple feature representation feedback.

4.3 Global Re-weighting

We are now ready to precisely describe the global re-weighting process that takes into account the contribution of both the relevant examples and the context images. In particular, we compute the global weight of the i -th coordinate as:

$$w_{new}[i] = \beta w^{\mathcal{R}}[i] + \delta w^{\mathcal{C}}[i] \quad (6)$$

where $w^{\mathcal{R}}[i]$ is the weight component computed on the current relevant examples following the re-weighting strategy proposed in [3], and $w^{\mathcal{C}}[i]$ represents the weight component derived from the context images (computed as described in the previous section). Finally, β and δ are weighting factors, satisfying the constraint $\beta + \delta = 1$.

As already mentioned at the beginning of Section 4, our re-weighting approach ensures an inertial behavior on the weights by means of the $w^{\mathcal{C}}[i]$ term. The evolution over time of \mathcal{C} , in fact, guarantees that relevant images that represent common (for the “near selection” strategy) or distinguishing (for the “distant selection” approach) characteristics of the relevant examples are taken into account for the re-computation of the weights in further search rounds.

5 Exploiting Prior Preferences

In the previous section, we have shown how the context-based approach to image similarity queries is able to complement available relevance feedback techniques, by providing a better starting point for the search of complex semantic concepts and by allowing an inertial behavior on the re-weighting process able to maintain the history of the “best” relevant examples. However, our approach share with all the relevance feedback methods the limit to “forget” user preferences across multiple query sessions. This means that the feedback loop has to be restarted for every new query. To overcome such limitation, many leaning user preferences techniques have been recently proposed (e.g., [1, 13, 15]) for implementing interacting similarity queries.

With particular attention to the **FeedbackBypass**⁵ technique presented in [1], the main idea is to store and maintain the information on the query parameters gathered from past feedback loops, in order to exploit it during new query sessions, either to bypass the feedback loop completely for already-seen queries, or to start the search from a near-optimal configuration for similar images. In other words, **FeedbackBypass** tries to “predict” what the user is looking for on the basis of the image query submitted to the system only. We synthetically represent this general approach as a mapping:

$$\mathbf{q} \mapsto (\mathbf{q}_{\text{opt}}, \mathbf{W}_{\text{opt}}) \quad (7)$$

which assigns to the initial image query \mathbf{q} an optimal query point \mathbf{q}_{opt} and an optimal set of weights \mathbf{W}_{opt} .

However, since **FeedbackBypass** only stores a single set of query parameters for each image query, the user cannot change her preferences to express a different semantics for a same query image. As also argued in [13, 15], this represents a limit because

⁵ **FeedbackBypass**: <http://www-db.deis.unibo.it/FeedbackBypass/>.

the assumption that the behavior of all users (or of the same user at different time) is the same for a given query is not always realistic (see also Figure 2).

We solve the above problem by means of our context-based approach. In details, we integrate the new query model $Q = (\mathbf{q}, \mathcal{C}, k)$ in the **FeedbackBypass** kernel, obtaining a new mapping definition:

$$(\mathbf{q}, \mathcal{C}) \mapsto (\mathbf{q}_{\text{opt}}, \mathbf{W}_{\text{opt}}) \quad (8)$$

which assigns to the couple $(\mathbf{q}, \mathcal{C})$ an optimal query point \mathbf{q}_{opt} and an optimal set of weights \mathbf{W}_{opt} . Thus, associating distinct contexts to the same image, it is now possible for **FeedbackBypass** to support different searches that use the same query.

In our current implementation of **FeedbackBypass** we use a *Support Vector Machine* (SVM) for regression [12, 2] to establish the mapping between $(\mathbf{q}, \mathcal{C})$ and the relative parameters $(\mathbf{q}_{\text{opt}}, \mathbf{W}_{\text{opt}}^{\mathcal{C}, \mathcal{R}})$. In fact, we experimentally found that the implementation of **FeedbackBypass** that uses SVM is more effective than the previous one based on Wavelets.

Given a set of training points, SVM for regression estimates the shape of the unknown function by selecting, among a set of a priori known functions, the one that minimizes the average error computed in approximating all the training points. As for the choice of the kernel for the SVM, we adopt the commonly used *Radial Basis Function* (RBF).

6 Experimental Evaluation

In this section we experimentally quantify the improvement introduced by our context-based approach in the similarity query process in term of effectiveness and efficiency. Results were obtained on a dataset of about 10,000 images extracted from the IMSI collection.⁶ Each image is represented in the HSV color space as a 32- D histogram, obtained by quantizing the Hue and Saturation components in 8 and 4 equally-spaced intervals, respectively.

The query workload consists of 10 randomly chosen images. For each query image, a set of volunteers defined a corresponding semantic concept (e.g., “animals in the forest”, “domestic animals”, “birds on the sea”, etc.) and images in the dataset were classified according to such “ground truth”: this allows the objective definition of relevance of an image wrt a query. Then, for each query, the users selected a context, i.e., a set of images able to contextualize the query with respect to its corresponding concept. In our current implementation, the context is defined by three images and at each step the context switch involves one image change.

To measure the effectiveness of our solution we consider the classical *precision* metric, i.e., the percentage of relevant images found by a query (in our experiments we set $k = 50$), averaged over the 10 queries. As for the efficiency, we compute the precision gain introduced by **FeedbackBypass** with respect to the results obtained without exploiting prior user preferences (at the same level of work, i.e., number of rounds). This is to show that **FeedbackBypass** indeed allows to reduce the number of search rounds needed to reach a given level of precision.

⁶ IMSI MasterPhotos 50,000: <http://www.imsisoft.com>.

The results we report refer to two main competitors:

1. **Default:** This is the approach currently used by all interactive retrieval systems, where the search is started by using the user query point and the default distance.
2. **Context:** Represents our context-based approach, where we search using a query point, a context and a distance function reflecting the meaning of the context images. In particular, depending on the strategy applied for the context switch and the context re-weighting, we specialize **Context** as follows:
 - (a) **Context-near:** applies the *near-selection* and the *maximum distance* policies;
 - (b) **Context-distant:** *distant-selection* and the *minimum distance* strategies are applied.

6.1 Experimental Results

Experiment 1. The aim of our first experiment is to measure the contribution of the context at the first round of search (i.e., without taking into account user preferences). To this end, we compare precision values obtained for the **Context** and the **Default** strategies, by considering both scenarios where **FeedbackBypass** is switched “off” (referred as **No FB**) and “on” (i.e., **FB**), respectively.

Results in Figure 3 confirm that **Context** consistently outperforms **Default** for the first search round. In particular, **Context** produces an average improvement over **Default** of 119.03% for the case **No FB** and of 24.13% for the case **FB**, respectively. The lower contribution for the latter case is due to the fact that here prior relevance judgements are exploited.

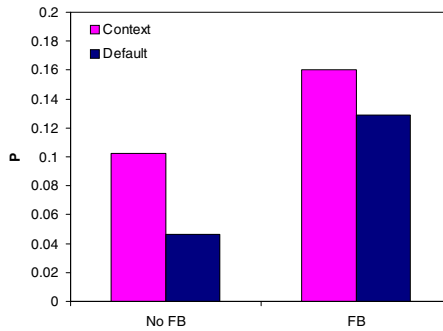


Fig. 3. Precision at the first search round for **Context** and **Default** strategies, without (**No FB**) and with (**FB**) **FeedbackBypass**

By analyzing the contribution of the **FeedbackBypass** technique in terms of search efficiency, we observe how the prior knowledge on user preferences is able to produce an improvement of precision of 176.19% for **Default** and of 56.52% for **Context**, respectively, with respect to the **No FB** scenario.

Experiment 2. In this second experiment our objective is to evaluate the precision dynamic (over 8 rounds) of **Context-near**, **Context-distant** and **Default** strategies (for

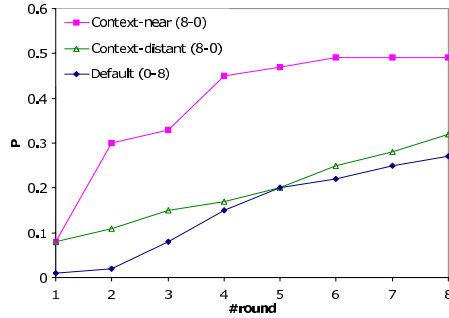


Fig. 4. Precision dynamics (8 rounds): Context-near and Context-distant strategies vs Default. The number pairs represent the number of rounds computed with and without context, respectively (e.g., (8-0): all 8 rounds executed using context).

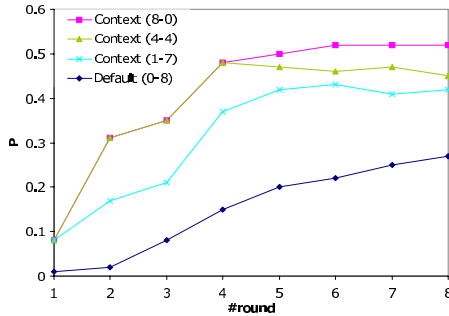


Fig. 5. Inertial behavior of the context measured at different rounds: Context vs Default

simplicity of explanation we only consider the case No FB). In this scenario, we make the assumption that user feedback is present. Results shown in Figure 4 report precision vs number of rounds of search and demonstrate that both Context-near and Context-distant strategies improve over Default of 312.5% and 87.5%, respectively, at the 3th round of search and of 81.48% and 18.51%, respectively, at the 8th step. It comes out that Context-near is the winning strategy, thus in the following experiments we will only focus on it. Its better effectiveness is due to the adopted near-selection approach that ensures an inertial behavior of the weights over time.

Experiment 3. The third experiment aims to quantify the influence of context at different search steps. To this end, we experimented with Context and drop the use of the context at a given search round (e.g., at 1st, 4th, or 8th round). We then compare precision results with those obtained by Default (0-8). Figure 5 shows that if we switch off the contribution of the context at the very beginning of the search (round 1) we obtain a final average improvement of 55.55% over Default. The improvement increases when we keep using the context, obtaining a 66.66% improvement at round 4 and a 92.59% at round 8, respectively. This shows that not only the use of the context increases the search effectiveness, but also that the inertial behavior of the context switch provides even better results.

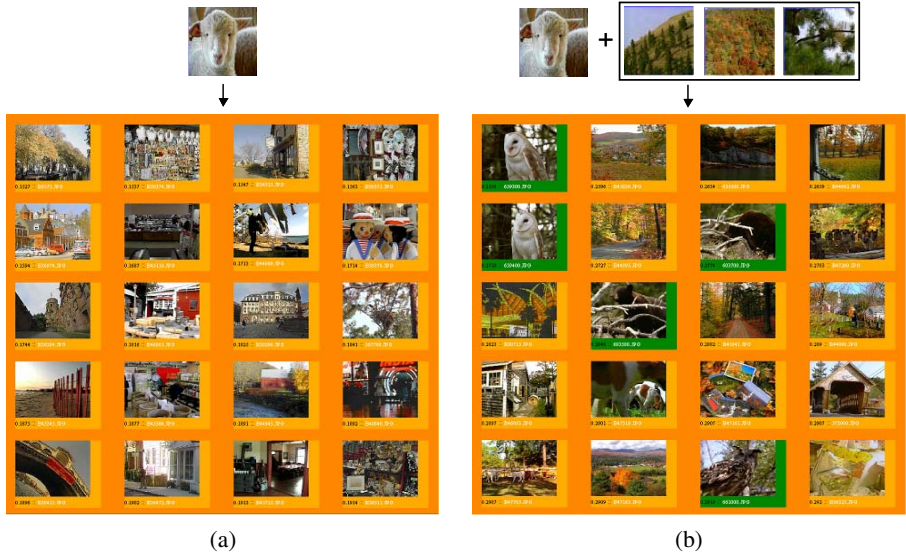


Fig. 6. Visual results at the first round of search for the semantic concept “animals in the forest”: Default (a) vs Context (b). Images in the same semantic class of the query are highlighted in green.

Visual Example. We conclude the section by showing actual query results obtained when using context. The aim is to provide evidence of the contribution of the context information with respect to the traditional approaches to similarity queries. In particular, in the example of Figure 6, showing the top 20 images for a query in the semantic class “animals in the forest”, Context returns 5 relevant images out of 20 (25% precision), whereas Default does not return any relevant object.

7 Conclusions

In this paper we have presented an approach to contextualize image queries, which is able to effectively represent complex semantic concepts by means of the notion of image context. Although this is simple, it is indeed effective and does not require neither a-priori classification of the image database, nor the analysis of surrounding text (e.g., image caption, text of Web page including the image, etc.), which might not always be available with an image. Furthermore, our approach easily complements available relevance feedback techniques, representing a “good” starting point for interactive searches, and helps increasing both the effectiveness and efficiency of further rounds of retrieval. This has been demonstrated through experiments on a dataset of about 10,000 manually classified images.

In this paper, for sake of definiteness, we have analyzed the performance of our method by considering the case where rather simple feature descriptors are used. Further work will include a thorough investigation of the effects of using more complex features and distance functions for similarity assessment.

References

1. I. Bartolini, P. Ciaccia, and F. Waas. FeedbackBypass: A New Approach to Interactive Similarity Query Processing. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 201–210, Rome, Italy, Sept. 2001.
2. H. Drucker, C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support Vector Regression Machines. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems 9*, pages 155–161, 1997.
3. Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Querying Databases Through Multiple Examples. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB'98)*, pages 218–227, New York, NY, USA, Aug. 1998.
4. M. Ortega and S. Mehrotra. *Relevance Feedback in Multimedia Databases*. In *Handbook of Video Databases: Design and Applications*. CRC Press, 2003.
5. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.
6. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.
7. S. Santini and R. Jain. Integrated Browsing and Querying for Image Databases. *IEEE MultiMedia*, 7(3):26–39, 2000.
8. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.
9. J. R. Smith and S.-F. Chang. VisualSEEK: A Fully Automated Content-Based Image Query System. In *ACM Multimedia*, pages 87–98, Boston, MA, Nov. 1996.
10. S. Smoliar and L. Wilcox. Indexing the Content of Multimedia Documents. In *Proceedings of the Second International Conference on Visual Information Systems (VISual'97)*, pages 53–60, San Diego, CA, Dec. 1997.
11. S. L. Tanimoto. *An Iconic Symbolic Data Structuring Schema*. In *Pattern Recognition and Artificial Intelligence*. Academic Press, N.Y., 1976.
12. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.
13. H. Wang, B. C. Ooi, and A. K. H. Tung. iSearch: Mining Retrieval History for Content-Based Image Retrieval. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA'03)*, pages 275–283, 2003.
14. T. Westerveld. Image Retrieval: Content Versus Context. In *Proceedings of the Content-Based Multimedia Information Access, RIAO, 2000*.
15. L. Zhang, F. Qian, M. Li, and H. Zhang. An Efficient Memorization Scheme for Relevance Feedback in Image Retrieval. In *Proceedings of the International Conference on Multimedia & Expo (ICME'03)*, 2003.

Information Retrieval of Sequential Data in Heterogeneous XML Databases

Eugen Popovici, Pierre-François Marteau, and Gildas Ménier

Valoria Laboratory, University of South-Brittany,
BP 573, 56017 Vannes Cedex, France
{Eugen.Popovici, Pierre-Francois.Marteau,
Gildas.Menier}@univ-ubs.fr

Abstract. The XML language is a W3C standard sustained by both the industry and the scientific community. Therefore, the available information annotated in XML keeps and will keep increasing in size. Furthermore, not only the volume of the XML information is increasing but also its complexity. The XML documents evolved from plain structured text representations, to documents having complex and heterogeneous structures and contents like sequential or time series data. In this article we introduce a retrieval scheme designed to manage sequential data in an XML context based on two levels of approximation: on the structural localization/organization of the sequential data and on its content. To this end we merge methods developed in two different research areas: XML information retrieval and sequence similarity search.

1 Introduction

The XML language is a W3C standard that has rapidly been adopted and sustained by both the industry and the research community. In the recent years, we witness at an increasing volume of XML digital information produced through day-to-day or by specialized scientific activities. Furthermore, not only the volume of the XML information is increasing but also its complexity. The XML documents evolved from plain structured text representations to documents having complex and heterogeneous structures and contents: multimedia description (MPEG7-DDL) and synchronization (SMIL), mathematical formulas (MathML), time series or sequences. We are mostly interested by the last two mentioned categories that are a ubiquitous form of data in financial, medical, scientific, musical or biological applications.

Flexible querying of scientific experimental results, patient's medical records, financial summaries, musical pieces or biological sequences published as XML documents are only a few examples of applications that involve managing sequential data in an XML context. We can thus state that there is a real need for high-performance systems and methods able to extract, index, and query heterogeneous types of sequential information from heterogeneous collections of XML documents.

In musical and biological fields special DTDs have been designed for midi files – MidiXML [1], musical scores – MusicXML [2] and biological sequential data [3] representation. In these cases the applications handle normalized sequential data and *data-centric* oriented XML documents. Data-centric documents have fairly regular structure, fine-grained data and little or no mixed content.

A heterogeneous collection of XML documents contains many un-normalized or various kinds of sequential data. This is more suited to a *document-centric* view of the database. Document-centric documents have less regular or irregular structure, larger grained data and lots of mixed content. Furthermore, the order in which sibling elements and PCDATA occurs is almost always significant.

Approximate matching in XML is closely related to the *document-centric* view and provides the possibility of querying the information acquired by a system having an incomplete or imprecise knowledge about both the structure and the content of the XML documents [4], [5], [6], [7].

One basic requirement of an XML query engine based on information retrieval concepts is to dispose of “vague predicates”/specialized similarity operators to adequately manage different data types [4] and to improve the precision of the IR system [8].

The approaches proposed in [4], [5], [6], [7] study the flexible querying on both XML structure and content (usually text), but do not specifically take into consideration sequential/time series data, nor its organization within the XML documents, which is the focus of our approach.

In our work we merge methods developed in two different research areas: XML information retrieval and sequence similarity search in order to provide adequate approximate operators for managing sequential/time series data in a heterogeneous XML environment.

In section 2 we introduce and formalize the underlying data model of our application and we identify relations between the structure of the XML documents and several common types of sequential data. In section 3 we present a hybrid indexing scheme allowing the implementation of semistructured and sequential searching operators. In section 4 we devise an approximate searching scheme for ranking the results by taking into account similarities between both the structural location and the content of the sequential data with the user requests. In Section 5 we conduct preliminary experiments dedicated to midi files retrieval embedded in heterogeneous XML databases. Finally, in Section 6 we summarize our conclusions and present some futures work perspectives.

2 Data Model

An XML document can be represented by an ordered tree whose nodes contain heterogeneous pieces of information (TEXT or PCDATA such as (parts of) sequences or time series). Each XML element may be related to attributes “name-value” fields, and each attribute value may contain (a part of a) sequential data.

Consider for example a phone number (a whole sequence) or a musical note (a sequence symbol) either as the content of an XML element node or as an XML attribute value.

To describe a sequence embedded in a heterogeneous XML environment we concurrently use its structural location in the collection – i.e. the set of XML contexts associated with the sequence symbols – and its content – i.e. the sequence symbols values.

2.1 XML Context

A Document Object Model (DOM) is an algorithmically structure that echoes the document organization in a graph, or tree. For an XML document, the DOM scheme is the tree of XML elements (as nodes) (Fig. 1 shows an example).

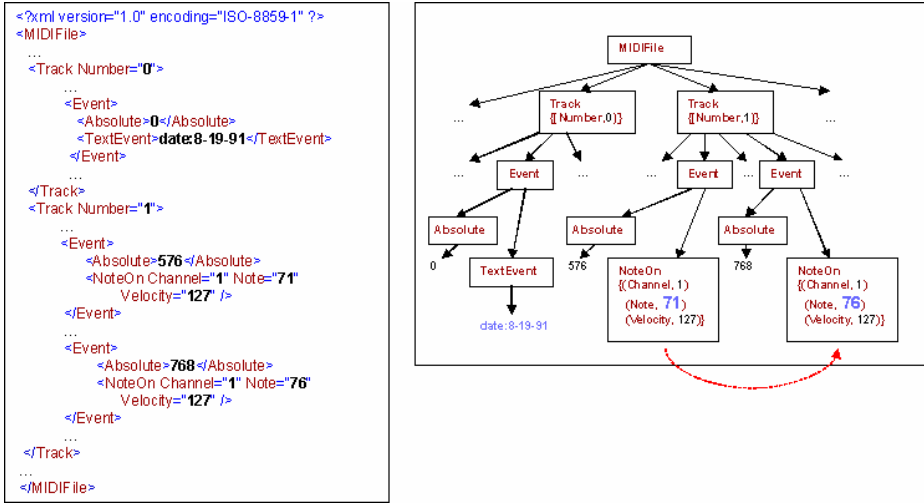


Fig. 1. An excerpt of an XML MIDI file [1] with its associated DOM tree

According to the tree structure, every node n of the DOM tree inherits a path $p(n)$ composed with the nodes that link the root to the node n . More precisely, $p(n)$ is an ordered sequence of nodes $p(n)=n_0n_1..n_i..n_dn$, where n_0 is the root node and $d+2$ the length of the sequence. An unordered set of $\langle attribute, value \rangle$ pairs $A(n_i)=\{ (a_j, v_j) \}$ may be attached to each node n_i of the ordered sequence, so that $p(n)$ can be represented as follows:

$$p(n)=\langle n_0, A(n_0) \rangle \langle n_1, A(n_1) \rangle \dots \langle n_n, A(n_n) \rangle \tag{1}$$

A node n in the DOM tree can be decomposed into structured and unstructured sub-elements. Moreover, an unstructured sub-element (*USE*) or an attribute value v_j may be decomposed into tokens t_i (or words, symbols, etc...). Each token t_i is related to an XML context $p(n)$ that characterizes its occurrence within the document.

2.2 Sequential Data

The XPath 1.0 Recommendation [9] defines the term *document order* as the order in which the first character of the XML representation of each node occurs in the XML representation of the document after the expansion of general entities, except for namespace and attribute nodes whose document order is application-dependent.

The XML document structure and the *document order* encode useful and potentially (semantically) rich information about the sequential organization of the

data. We describe and formalize hereinafter our approach in exploiting this kind of information for XML sequence extraction and representation.

Sequence Definition. Formally, a sequence $S = s_0 s_1 \dots s_i \dots s_m$, is defined relatively to a collection of XML documents as an ordered and finite non-empty set of symbols $\{s_i\}$ selected from an alphabet Ω . An alphabet symbol s_i may be represented by:

- $v_j \in A(n_i)$ an attribute value,
- $t_i \in v_j \in A(n_i)$ a token composing an attribute value,
- USE_i an unstructured sub-element¹ of one of the XML nodes n ,
- $t_i \in USE_i$ a token of an unstructured sub-elements USE_i .

In the example of Fig. 1., a symbol value may indicate the “0” value of the *Number* attribute – for an attribute value type – or the *TextEvent* content “date: 8-19-91” for an *USE* type.

A sequence symbol s_i is linked with one of the unique indexed reference locators $rl_0 rl_1 \dots rl_i \dots rl_n$ of the XML collection set. A reference locator rl_i (defined in Section 3.1) points to a unique position in the collection of XML documents and includes a reference to the symbol’s XML context $p(n)$.

Two symbols associated to the same XML context $p(n)$ could be associated: one being considered to be an order key o_i (e.g. a timestamp), the other a sequence symbol s_i .

Sequence Structural Types. The above sequence definition allows representing sequences of symbols associated with any arbitrary XML contexts from the collection. From a more practical point of view, several particular structural types of sequences frequently occur and prove to be of interest:

- *node level sequence*: the whole sequence is contained in a single node of the DOM tree – as an *USE* or an attribute value –, usually a leaf. This sequence representation is widely used in bioinformatics [3],
- *document level sequence*: composed by the symbols associated to the approximate matched XML contexts of a single XML document – i.e. this includes perfect paths matches (e.g. see the link between the two nodes of the DOM tree from Fig. 1.), sibling nodes and nodes having a k-level common ancestor. This representation is imposed by the DTD’s used in the musical field [1], [2],
- *collection level sequence*: composed by the symbols associated to the approximate matched XML contexts by crossing the physical boundaries of the XML documents. This sequence type may be of interest when searching sequences of information spread among several documents and that are not entirely dependent of the *document order* – e.g. time stamped information selected by the social security ID number from medical records databases.

One of the main advantages of the first two sequence types is the possibility of maintaining the *document order* by default when two ordering keys o_i and o_j , with $i \neq j$, have identical values or the sequence order relation \leq is either:

¹ In the case of a node having a mixed content, only the unstructured content is considered.

- unspecified (no valid ordering key o_i has been extracted and associated with the sequence symbols s_i), or
- a partial order (indeterminate for certain cases, like the relationship between the temporal information: durations, dateTime, etc. as defined in XML Schema Part 2: Datatypes Recommendation [10]).

For sequences constructed with symbols extracted from different XML documents, the *document order* could be locally applied within each XML document, but no global order of the symbols in the sequences can be inferred without the use of external information (i.e. user provided order keys) and/or heuristics. A simple example will be the use of the document creation date for ordering the documents. We may also consider using the timestamp information associated with the symbols closest ancestors in the XML trees. In our scheme we make no assumptions about the global order of the symbols extracted from different XML documents.

Thus, we propose a model in which the symbols are organized in sequences by taking into account: (1) similarities between their structural positions in the XML document trees (i.e. using $p(n)$), (2) type compatibility between their values (numbers, dates or strings) and (3) an order relation \leq . A symbol may occur in more than one sequence and a sequence may contain symbols with identical values.

Sequence Extraction. The users' interests in a heterogeneous XML environment can be highly diversified. Some could search the chorus of a musical piece or, in another case, similarities between the blood pressures curves of several patients' medical records. In these conditions we will probably fail to index all the different sequences that could match the users' subjective and time evolving interests.

We assume that the users or the system administrators detain at least an imprecise, incomplete or fuzzy knowledge of the particular underlying organization of the sequential data in which they are interested in. This assumption makes credible the fact that they will be able to supervise the sequential data extraction process conforming to their specific needs.

The Sequence Extraction Process. The sequence extraction process is based on an construction operator *makeSeq* that receives three arguments:

- an XML context $p(n)$ with the type and position of the requested symbol s and (optionally) of the order key o ,
- the minimum accepted threshold for the match of the symbols location with the provided XML context (see Section 4.1) in order to be included in the current sequence, and
- the sequence expected type: node, document (default value), or collection.

During the extraction process, the compatibility constraints expressed on the symbols values are mandatory while the structural location of the symbols and their order relation are treated as approximate. Therefore applying the *makeSeq* operator to the input XML data – i.e. to the collection of XML contexts – result in a set of symbols s that are associated to XML contexts similar with the one provided by the user.

In this phase of the process, the sequences are built as indicated by the sequence expected type parameter and are eventually ordered by using the symbols order keys

o_i . In the case of a mixture of symbols associated or not to an order key within the same sequence, the symbols without an order key are discarded.

3 Indexing Scheme

We propose a hybrid index model (Fig. 2) designed to merge both types of data: semistructured and sequential data.

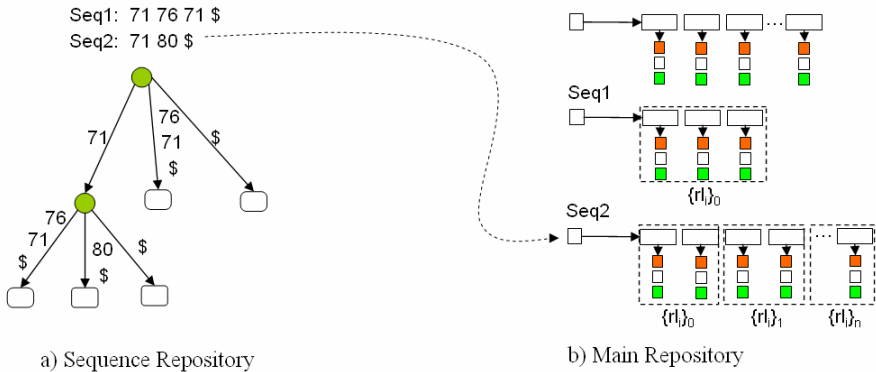


Fig. 2. The index model

3.1 Main Repository

The main repository uses inverted lists as basic structures. For this model, the entries of the inverted lists are of three kinds:

- structural entries, i.e. nodes n of the DOM tree,
- tokens of the unstructured sub-elements of the DOM tree nodes $t_i \in USE_i \in n$,
- sequential entries, i.e. unique sequence IDs. – *USIDs*.

Each entry has associated a list of reference locators rl_i pointing to a unique position in the collection of XML documents. A reference locator rl_i has attached three pieces of information:

- a link to the URI of the document,
- a link toward the XML context $p(n)$, and
- an index specifying the location of the token within the DOM node n .

For the structural entries (i.e a node n of the DOM tree), only the link to the URI of the document and a link to the XML context $p(n)$ are required.

The inverted list resulting from the indexing process is encoded using binary randomized search trees, namely TREAPS as defined in [11], associated to hashtable structures. TREAP structures can balance the search tree according to the frequency of requested items. This provides an access speed-up over the use of regular hashtables [12]. The inverted lists are implemented as disk-resident index structures.

3.2 Sequence Repository

A supplementary index structure is used to optimize the management of sequential data. A well known and efficient data structure for indexing and searching sequential data in text processing [13] and bioinformatics applications [14], [15] is the generalized suffix tree (GST). In a nutshell, the suffix tree is an indexing structure for all the suffixes of a string and it can be constructed in linear time and linear space [16]. A generalized suffix tree is a suffix tree for representing all the suffixes of a set of strings [17]. In our implementation we use the Ukkonen's on-line construction algorithm [18] for building a GST.

For each extracted sequence S_i , all its symbols values s_i are indexed in the generalized suffix tree structure with respect to the sequence order relation \leq .

The same sequence of indexed values S_i can have associated multiple sets of reference locators $\{rl_i\}_o\{rl_i\}_1\dots\{rl_i\}_n$ as it may occurs in different locations in the collection of XML documents. We consider a sequence to occur in two different locations in the XML collection if: (1) all its symbols values s_i are equal with the ones of an already indexed sequence and (2) at least one of their symbols associated reference locators rl_i are not matching. Otherwise, the sequence is considered to be a true duplicate and thus, discarded.

Consequently, for each indexed sequence we receive a unique sequence id $USID$, from the generalized suffix tree. The $USID$ is further used as an entry key in the inverted lists to link the multiple sets of reference locators to the referred sequence. The sets of reference locators have all the same cardinality $|S_i|$ and are stored contiguously in the inverted list. The order of the reference locators from each set complies with the order relation \leq used to build the sequence.

4 Searching Scheme

We introduce a searching scheme designed to manage unstructured sequential/time series data in an XML context based on two levels of approximation: on the structural localization/organization of the sequential data and on its content.

4.1 Structure Approximate Matching

As the user cannot be aware of the complete DOM structure of the database due to its heterogeneity, efficient searching should involved exact and approximate search mechanisms. The format of the indexed documents being XML, it is natural to consider that the structural query itself complies, at least partially, with the XML standard. If so, the structural searching mechanism can be handled through approximate tree matching algorithms that try to match the query DOM tree to the DOM trees for the corresponding indexed documents.

Tree matching algorithms exist based on editing distance and mappings – see [19] and [20] for instance. The complexity of the matching of two trees T_1 and T_2 is at least for [19] $O(|T_1|.|T_2|)$, where $|T_i|$ is the number of nodes of tree T_i . The complexity is much higher for common subtree search [20].

This kind of tree matching is not suited to the task we intend to perform. First of all, the matching complexity is too high considering the size of documents and data

bases we want to handle. Secondly a sequence may involve paths of more than one DOM tree of the collection. Therefore we have chosen to perform an approximate search based on the matching of $p(n)$ sub-structures of indexed DOM trees [12]. In other words, we are rather dealing with approximate root-leaf or root-node path alignment rather than complete tree matching algorithm.

In particular, a sequence S is defined as an ordered non-empty set of finite symbols each of them having attached a $p(n)$ sub-structure of the XML collection T_i^D DOM trees. The ordered set of $p(n)$ sub-structures of the DOM trees represents the structural part of the sequence, while the symbol values its content. Thus, approximate matching the structural part of an indexed sequence is based on T_i^D approximate $p(n)$ path alignments.

Approximate Matching of $p(n)$ Sub-structures. For that purpose, we evaluate the similarity between a root-node path p^R expressing an elementary structural query and T_i^D the DOM tree(s) corresponding to an indexed document/or sequence S as follows:

$$\delta(p^R, T_i^D) = \text{Min}_i \delta_L(p^R, p_i^D) \quad (2)$$

where δ_L is a Levenshtein type distance [21] and $\{p_i^D\}$ the set of root-node paths for document DOM tree or sequence S .

The complexity of such an algorithm – see [22] for justification – is :

$$O(\text{length}(p^R) \cdot \text{depth}(T^D) \cdot |\{p_i^D\}|) \quad (3)$$

where $|\{p_i^D\}|$, stands for the cardinality of the set $\{p_i^D\}$. For documents or sequences associated with DOM tree(s) having a small depth this complexity is perfectly tractable, even when the number of documents is high as far as the number of leaves is reasonable. Nevertheless, for most of day-to-day documents, equivalent DOM trees will exhibit a very low depth compared to their rather huge width.

Path Similarity Computation $\delta(p^R, T^D)$. Let p^R be the path for the structural request R and $\{p_i^D\}$ the set of root-leave paths of the DOM tree(s) associated to an index document or sequence S .

We designed an editing pseudo-distance [12] using a customised cost matrix to compute the match between a path p_i^D and the request path p^R . This scheme, also known as modified Levenstein distance, computes a minimal sequence of elementary transformation to get from p_i^D to p^R . The elementary transformations are:

- **Substitution:** a node n in p_i^D is replaced by a node n' for a cost $C_{subst}(n, n')$. Since a node n not only stands for an XML element, but also for attributes or attributes relations, we compute $C_{subst}(n, n')$ as follows:

1. $n.\text{element} \neq n'.\text{element}$: $C_{subst}(n, n') = 2$
(full substitution)
2. $n.\text{element} == n'.\text{element}$:
 - if $n'.\text{attributesCond}(n.\text{attributes})$ is true
 $C_{subst}(n, n') = 0$ (no substitution)
 - else $C_{subst}(n, n') = 1$ (only attribute substitution)

where `attributesCond` stands for a condition (stated in the request) that should apply to the attributes (for example the value of attribute `Channel` for the `NoteOn` element should be equal to “I”),

- **Deletion:** a node n in p_i^D is deleted for a cost $C_{del}(n)(=2)$,
- **Insertion:** a node n is inserted in p_i^D for a cost $C_{ins}(n)(=2)$.

For a sequence $Seq(p_i^D, p^R)$ of operations, the global cost $GC(Seq(p_i^D, p^R))$ is computed as the sum of the costs of elementary operations.

The Wagner&Fisher Algorithm [22] computes the best $Seq(p_i^D, p^R)$ (i.e. minimizes $GC()$ cost) with a complexity of :

$$O(\text{length}(p_i^D) * \text{length}(p^R)) \text{ as stated earlier.} \tag{4}$$

We postulate this scheme adequate for this application because of the limited depth of the XML DOM (mostly < 20) [23]. Let

$$GC_{min}(p_i^D, p^R) = \text{Min}_k GC(Seq_k(p_i^D, p^R)) \tag{5}$$

Given p_i^D and p^R , the highest possible value for GC can be evaluated to:

$$GC_{max}(p_i^D, p^R) = C_{subst} * (\min(\text{length}(p_i^D), \text{length}(p^R))) + C_{ins} * (|\text{length}(p_i^D) - \text{length}(p^R)|), \tag{6}$$

which can be interpreted as: all the XML elements are different, leading the GC to $\min(\text{length}(p_i^D), \text{length}(p^R))$ substitutions and $|\text{length}(p_i^D) - \text{length}(p^R)|$ insertions: this value of GC can be interpreted as no possible match (or no acceptable match). Here we have $C_{subst}=2$ and $C_{ins}=2$.

We postulate a ‘yet’ acceptable match: a perfect match for the XML elements, and no match for the attributes:

$$GC_{mid}(p_i^D, p^R) = C_{subst} * (\min(\text{length}(p_i^D), \text{length}(p^R))). \tag{7}$$

That is: the paths p_i^D and p^R have the same number of nodes, the same XML elements, but the attributes in p_i^D do not match the conditions in p^R . Here, we have $C_{subst}=1$ (attribute substitution only).

And, of course, $GC_{perfect}(p_i^D, p^R) = 0$ for a perfect match (same XML Elements, true conditions for the attributes).

The distance $d(p_i^D, p^R)$ is then computed mapping the GC_{min} value on the $[0,1]$ interval (Fig. 3).

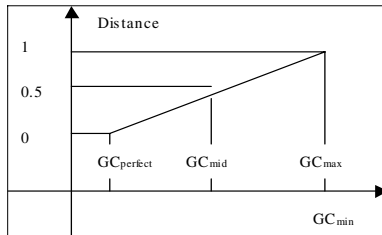


Fig. 3. Distance normalization

A structural similarity value is computed as follows:

$$Struct_{sim}(p_i^D, p^R) = 1 - d(p_i^D, p^R). \quad (8)$$

For a whole document D or sequence S , and a structural requests R :

$$Struct_{Matching}(D|S, R) = \underset{i}{Max} (Struct_{sim}(p_i^D, p^R)) \quad (9)$$

Thus, the documents/sequences which lead to the higher structural matching value are the best document/sequences matching candidate for a request p^R . Note that the matching value belongs to $[0,1]$, ranging from 0 (not acceptable – no matching), 0.5 (yet acceptable) to 1 (perfect matching).

4.2 Sequence Approximate Matching

We presume that a user has an imprecise, incomplete, or approximate knowledge of the sequential content extracted from the collection of XML documents. As a direct consequence, a sequential query S_q will represent only a short, probably inaccurate, fragment of an indexed sequence. These types of queries are usually found in query by humming systems [24] or in biological data processing [14],[15], [17]. Under these conditions, an exact searching scheme for sequence retrieval will fail to responds to the user information needs. Thus, we choose a retrieval scheme that approximately matches a sequential query S_q with any subsequence S_i^j of the indexed data.

The sequence similarity is based on a distance δ obtained by applying a dynamic programming technique: an editing Levenshtein distance [21] or Dynamic Time Warping (DTW) distance² [25]. Both distances have the same computational complexities $O(|S_i||S_q|)$ and allow the approximate matching of two sequences or time series of different lengths.

We want to retrieve all the similar subsequences S_i^j from the database with a user query S_q , having the distance δ less than a specified threshold \mathcal{E} - the *P-against-all problem* [17]. The sequential scan complexity for achieving this goal is expressed as:

$$O\left(m \overline{|S_i|}^2 \cdot |S_q|\right), \quad (10)$$

where m is the number of data sequences whose average length is $\overline{|S_i|}$. The use of a hybrid method based on a suffix tree as an index structure and a dynamic programming method can reduce the problem complexity and efficiently retrieve similar subsequences [14], [17], [25]. The performance gain of the method comes from (1) the branch-pruning method that reduces the research space using the threshold value ε and (2) the suffixes with common prefixes that share the cumulative distance tables during the index traversal. The time complexity of this approach is:

$$O\left(\frac{m \overline{|S_i|}^2 |S_q|}{R_d R_p}\right), \quad (11)$$

² DTW is a pseudo-distance as it is not respecting the triangular inequality, see [26] for demonstration.

where $R_d (\geq 1)$ is the reduction factor saved by sharing the cumulative distance tables, and the $R_p (\geq 1)$ is the reduction factor gained from the branch-pruning. In the worst case where there is no common subsequence and the branch-pruning cannot help, both values of R_d and R_p are 1, and therefore the complexity becomes the same as that of the sequential scan [25].

The similarity between a sequential query S_q and an indexed subsequence S_i^j is given by their normalized distance δ and is computed as follows:

$$Seq_{Sim}(S_q, S_i^j) = b^{-\delta(S_q, S_i^j)}, \tag{12}$$

where $b > 1$, usually e and $Seq_{Sim}(S_q, S_i^j) \in [0..1]$.

The value of the b parameter sets the sensitivity of the sequence similarity indicator. It specifies the distribution of the possible distance values δ in the $[0..1]$ interval and boosts the best matches. The sequence similarity takes values between 0 - for no correspondence between sequences, and 1 - for a perfect matching.

The match of a sequential query S_q with an index sequence S_i is defined as the best match between the query and any subsequence S_i^j of S_i :

$$Seq_{Matching}(S_q, S_i) = \underset{j}{Max}(Seq_{Sim}(S_q, S_i^j)), \tag{13}$$

4.3 The Fusion of Structure and Sequence Approximate Matching Scores

The *information fusion* is defined as the fusion of complementary information provided by different sources with the scope of obtaining an information gain due to the utilization of multiple sources of information vs. a single source.

In our scheme we have chosen the weighted geometric mean to fuse the two levels of approximation: on the structural localization/organization of the sequential data and on its content. The geometric mean is a way to construct an aggregate measure between different indicators that is sensitive to small values. This is appropriate for our purpose of retrieving sequences with highly similar content and being related to highly relevant structures to the user query. The weighted geometric mean is defined as:

$$\Phi(Seq_{Matching}, Struct_{Matching}) = \alpha_1 + \alpha_2 \sqrt[\alpha_1]{Seq_{Matching}^{\alpha_2} \cdot Struct_{Matching}^{\alpha_1}}, \tag{14}$$

where $\alpha_1/\alpha_2 = \lambda$ is a parameter allowing to specify the relative importance of the indicators to the final score.

We rewrite the above formula in order to transform the logarithmic scale of the λ parameter to a linear scale that is more suited to the user common-sense understanding:

$$\Phi(Seq_{Matching}, Struct_{Matching}) = 1 + \lambda \sqrt[1 + \lambda]{Seq_{Matching} \cdot Struct_{Matching}^\lambda}, \tag{15}$$

where $\lambda = -\log_2(1 - \gamma)$ and $\gamma \in [0..1]$.

The γ parameter is application dependent and it is used for specifying the degree of penalty applied to the final score with respect to the structural matching indicator. A $\gamma=0$ value will discard the sequence structural factor from the calculus of the overall score, while a $\gamma \rightarrow 1$ value will boost its importance at maximum. At $\gamma=0.5$ the fusion process will equally take in consideration the two indicators.

5 Experimental Results

We present some preliminary experimental results dedicated to midi files retrieval in a heterogeneous XML MIDI library.

We have implemented the approximate sequential matching operators and the fusion method based on the presented index model in the SIRIUS XML Information Retrieval Engine [12]. The prototype is entirely developed in Java and uses the Dynamic Time Warping [25] to compute the similarity between sequences. The implementation of the similarity search for sequence retrieval follows the algorithms introduced in [25].

5.1 Experimental Dataset

The experimental dataset is formed by a MIDI file collection (32 Disney themes) downloaded from the public domain³. The files are transformed⁴ in the XML format conforming to the Standard MIDI File DTD [1] version 0.9.

In a MIDI file the sequences of notes are organized in tracks (maximum 16 channels) and events (see Fig. 1).

To simulate the heterogeneity of the collection and to validate the approximate structural localisation of the sequential data, we randomly generate and append a meta-structure to each standard XML midi file.

5.2 Early Evaluations

We present some early experiments of the XML data indexing and sequence extraction algorithms on datasets with sizes ranging from 1MB to 15MB. The system used for experimentations disposes of a 2.4 GHz processor and 512 RAM. The xml data indexing time represents the elapsed time for the creation of the inverted lists without taking into consideration the sequential data. The sequence extraction time stands for the time spent in the process of approximate matching the XML contexts and the time spent to index the extracted sequences.

We can observe in Fig. 4 quasi linear indexing and extraction times with the size of the indexed datasets (i.e. the total length of the indexed sequences), which is quite encouraging. The average response time for 90 randomly generated requests are shown in Fig.5. The structural requests seems to have a polynomial behavior. The ones due to the organization of the GST index structure. A GST scales well to the sequential queries are less sensitive to the size of the indexed data than the structural dataset size as it uses the common prefixes of the index sequences to reduce the research space (see section 4.2).

³ <http://themes.mididb.com/anthems/>

⁴ <http://staff.dasdeck.de/valentin/midi/>

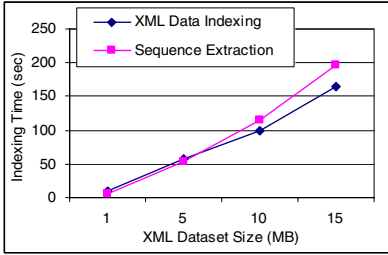


Fig. 4. XML Indexing / Sequence extraction time as the size of the index datasets

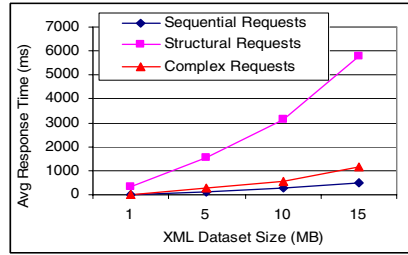


Fig. 5. Average response time for structural, sequential, and complex requests

This fact raises interesting perspectives for the optimization of the overall response time of the complex requests involving both the structure and the sequential content of the XML documents. We use the sequential queries as a first filter when answering to complex requests. This improves significantly the overall response time as shown in Fig. 5. Finally, the complexity of the alignment algorithms is maintained as low as possible to preserve the capability of indexing large data sets.

Considering the quality of the retrieved results, we could not evaluate it completely, as an evaluation framework for the retrieval of multimedia structured document fragments [27] is still under development at the moment of writing this article. A general opinion [8] and also our belief is that using similarity operators adapted to the document content types and to the XML structure in the retrieval process will improve the precision of the results.

6 Conclusion

We have described approximate searching mechanisms to retrieve and query sequential data from semi-structured information embedded into XML data sets. Such mechanisms are based on the alignment of root-node paths that are sub-structures of XML DOM trees. The proposed mechanisms allow to fusion structured data (*<attribute, value>* pairs) or structural organization of documents (*<MIDIFile> <Track> <Event> <NoteOn>...*) with unstructured data such as textual (free text) or sequential/time series data.

At the current authors’ knowledge, there is no existing integrated method for approximate querying specific sequential data in a heterogeneous semi structured environment. Even if each part of the problem have been extensively studied and have beneficiated of strong research efforts of well established scientific communities, the fusion of the methods developed in this two research areas (sequential similarity search and XML information retrieval) was not yet deeply considered. The proposed scheme was designed in order to cover this gap and to highlight extended and useful querying capabilities for the final user.

Our main experimental contribution so far, shows that the fusion of structural and sequential search criteria could drastically improve the response time as well as the retrieval performances of the similarity search mechanisms when exploiting heterogeneous XML databases.

We intend to explore and enlarge the set of the sequential operators implemented in the system by making them aware of the temporal aspects of the data and by allowing a flexible ordering of the symbols composing the sequences. Both, the disk resident organization of the index structures and the parallelization of the research algorithms will be a straight forward research direction in order to validate our approach on important volumes of data.

Acknowledgements

This work was partially supported by the ACIMD – ReMIX (Reconfigurable Memory for Indexing huge amount of data).

References

1. MidiXML, Standard MIDI File DTD: MIDI XML, Version 1.0 - 13 January 2004, <http://www.recordare.com/dtts/midixml.html>, (2004)
2. MusicXML, MusicXML Definition, Version 1.0, January 2004, <http://www.recordare.com/xml.html>, (2004)
3. Robinson A., "XML's and DTD's for Biology", An XML Workshop for Biologists and Bioinformaticians, <http://industry.ebi.ac.uk/~alan/XMLWorkshop/>, (2000)
4. Fuhr N., Großjohann K., XIRQL: An XML query language based on information retrieval concepts, *ACM Transactions on Information Systems (TOIS)*, v.22 n.2, p.313-356, April 2004
5. Amer-Yahia S., Koudas N., Srivastava D., Approximate Matching in XML, *Advanced Technology Seminar 5, ICDE 2003*
6. Amer-Yahia S., Laks V. S. Lakshmanan, Shashank Pandit, FlexXPath: Flexible Structure and Full-Text Querying for XML. *SIGMOD Conference, Paris France, June 2004*, pp. 83-94.
7. Carmel D., Maarek Y. S., Mandelbrod M., Mass Y. and Soffer A., Searching XML documents via XML fragments, *SIGIR 2003, Toronto, Canada* pp. 151-158.
8. Dorneles C. F. , Heuser C. A. , Lima A. E. N., Da Silva A., De Moura E., "Measuring similarity between collection of values", *6th ACM International Workshop on Web Information and Data Management, WIDM (2004)*
9. Clark J., DeRose S., XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999, <http://www.w3.org/TR/xpath.html>, (1999)
10. Biron P., Malhotra A., XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>, (2004)
11. Seidel R., Aragon C. R., "Randomized Binary Search Trees", *ALGORITHMICA*, 16(4/5):464-497, (1996)
12. Ménier G., Marteau P.F., Information retrieval in heterogeneous XML knowledge bases, *The 9th International Conference on Information Processing and Magement of Uncertainty in Knowledge-Based Systems , IEEE*, 1-5 July, (2002), Annecy, France.
13. Navarro G., A Guided Tour to Approximate String Matching, *ACM Computing Surveys*, Vol. 33, No. 1, p. 31-88, March 2001
14. Meek C., Patel J.M., and Kasetty S., Oasis: An online and accurate technique for local-alignment searches on biological sequences. In *Proc. 2003 Int. Conf. Very Large Data Bases (VLDB'03)*, pages 910-921, Berlin, Germany, Sept. 2003
15. Hunt E., Atkinson M.P., Irving R.W., Database indexing for large DNA and protein sequence collections, *The VLDB Journal*, Vol. 11 , n. 3, p.256 - 271, (November 2002)

16. McCreight E.M., "A Space-Economical Suffix Tree Construction Algorithm", *Journal of the ACM*, 23:262-272, (1976).
17. Gusfield D., *Algorithms on strings, trees and sequences*, Cambridge University Press, (1997)
18. Ukkonen E., "On-line construction of suffix-trees", *ALGORITHMICA*, Vol. 14, 249-260, (1995)
19. Tai, K.C., "The tree to tree correction problem", *J.ACM*, 26(3):422-433, (1979)
20. Wang T.L.J, Shapiro B., Shasha D., Zhang K., Currey K.M., "An algorithm for finding the largest approximately common substructures of two trees", In *J. IEEE Pattern Analysis and Machine Intelligence*, vol.20, N 8, August (1998)
21. Levenshtein A., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", *Sov.Phys. Dohl.* Vol.10, P.707-710, (1966)
22. Wagner R., Fisher M., "The String-to-String Correction Problem", *Journal of the Association for Computing Machinery*, Vol.12, No.1, p.168-173, (1974)
23. Mignet L.,Barbosa D.,Veltri P., *The Web XML: A First Study* (2003), <http://citeseer.ist.psu.edu/mignet03web.html>
24. Zhu Y., Shasha D., *Warping indexes with envelope transforms for query by humming*, *Proceedings of the 2003 ACM SIGMOD*, San Diego, California, p. 181 – 192.
25. Park S., Chu W., Yoon J., Won J., "Similarity search of time-warped subsequences via a suffix tree", *Information Systems*, 28(7): 867-883, (2003)
26. Yi B., Jagadish H. V., Faloutsos C., "Efficient Retrieval of Similar Time Sequences Under Time Warping", *ICDE*, pp. 201-208, 1998.
27. R. van Zwol, G. Kazai, M. Lalmas, *Multimedia track, INEX*, April 2005 - December 2005, <http://inex.is.informatik.uni-duisburg.de/2005/tracks/media/index.html>

A Visual Annotation Framework Using Common-Sensical and Linguistic Relationships for Semantic Media Retrieval

Bageshree Shevade and Hari Sundaram

Arizona State University
{Bageshree.Shevade, Hari.Sundaram}@asu.edu

Abstract. In this paper, we present a novel image annotation approach with an emphasis on – (a) common sense based semantic propagation, (b) visual annotation interfaces and (c) novel evaluation schemes.

The annotation system is interactive, intuitive and real-time. We attempt to propagate semantics of the annotations, by using WordNet and ConceptNet, and low-level features extracted from the images. We introduce novel semantic dissimilarity measures, and propagation frameworks. We develop a novel visual annotation interface that allows a user to group images by creating *visual* concepts using direct manipulation metaphors without manual annotation. We also develop a new evaluation technique for annotation that is based on relationship between concepts based on commonsensical relationships. Our Experimental results on three different datasets, indicate that the annotation system performs very well. The semantic propagation results are good – we converge close to the semantics of the image by annotating a small number (~16.8%) of database images.

1 Introduction

In our work, annotation is considered to be the meta-data or the keyword description that is given to images. For example – if you have an image of a cat, then the keywords “*cat*”, or “*animal*” etc. that users would give to describe that image is considered to be annotation. The goal of our work is to create novel semi-automated, intelligent annotation algorithms that bridge manual methods for annotation and fully automatic techniques. Though fully automatic techniques based on pattern recognition and content based retrieval are efficient, they are not very accurate. On the other hand, manual techniques though accurate are very tedious and time consuming.

There has been prior work in semi-automatic image annotation using relevance feedback [2,3,8,11,13]. While there are rich mathematical models used, we believe that there are three shortcomings of the current work:

- **Semantics:** Current approaches to annotate images [2,3,11,13] essentially treat words as symbols regardless of their semantic relationships with other words, which is no different than any normal image feature. The lexical meaning of the keywords/annotations is not exploited.

- **Intuitive Interfaces:** There are number of tools and drag and drop interfaces [5,11] for annotating images. However, these tools do not allow users to group images based on *visual* concepts without the use of manual annotations. Current annotation tools also lack any kind of label propagation.
- **Novel Evaluation Schemes:** Presently, all CBIR systems [2,3,8,13] predominantly use the Precision-Recall metric, to evaluate their system. However, this measure does not take into account the semantic relationship (e.g. though linguistic ontologies) among words. Hence, there is a need to develop new evaluation techniques that incorporate semantics.

In this work we address issues relating to both semantics and visual annotation interfaces. We establish the semantic inter-relationships amongst the annotations by using WordNet [7] and ConceptNet [6]. We also develop an intuitive visual annotation interface.

The annotation procedure is as follows. Our annotation system uses a combination of low-level features such as color, texture and edge as well as WordNet synsets and ConceptNet distances to propagate semantics. As the user begins to annotate the images, the system creates positive example (or negative examples) image sets for the associated WordNet meanings. These are then propagated to the entire database, using low-level features as well as ConceptNet distances. The system then determines the image that is least likely to have been annotated correctly and presents the image to the user for relevance feedback.

Our annotation interface allows users to group images by creating *visual* concepts without requiring text annotation. A visual concept is an abstract idea that the user associates with an image. For example – the user could associate the concept of “*garden*” to a group of semantically related images depicting flowers. The user can then add positive (or negative) examples to these visual concepts. The system then creates visual clusters based on low-level features. The user can also add text annotations to these visual concepts. The system then propagates these annotations to all the visual concept clusters based on low-level features and WordNet.

The annotation system is evaluated using a novel evaluation scheme that is not based on the Precision-Recall metric. Our system uses ConceptNet similarity measure, to determine the performance of our annotation algorithm. This is done since the Precision-Recall measure does not incorporate the semantic relationship between words. Traditional precision-recall measures are useful in the context of classification. However, in annotation, the semantics of the annotation words are important. For example – if the system associates the word “*apartment*” to an image that is annotated as “*house*”, then it is not such a large error. We therefore, use semantic relationship between words to evaluate the accuracy of the annotation algorithm. Our results indicate that the system performs very well and is much better than the baseline case of binary concept membership.

The rest of this paper is organized as follows. In the next section, we discuss the features used in our system. In section 3, we present briefly the semantic propagation algorithm. In Section 4 we present a visual annotation interface while in Section 5 we present the experimental results. And finally, we present our conclusions in Section 6.

2 Features

In this section, we shall describe the low-level features as well as the semantic features used in our annotation system.

2.1 Media Features

In our work, the feature vector for images comprises of color, texture and edge histograms. The color histogram consists of 166 bins in HSV space. The HSV space is used, as it is perceptually continuous. The system extracts Tamura texture [4] from images. The texture histogram consists of 3 bins corresponding to contrast, coarseness and directionality of the image. The edge histogram [4] consists of 71 bins that incorporates curvature and edge directionality. We then concatenate the three histograms to get a final composite histogram of 240 bins. The low level feature distance between two images i and j is then given as:

$$d(i, j) = \sqrt{\sum_{k=1}^N (h_i^k - h_j^k)^2}, \quad (1)$$

where N is the total number of bins, and h_i and h_j are the corresponding bins of images i and j .

2.2 Media Semantics

In our prior work [9], semantics were incorporated through the use of WordNet ontology, which is an online lexical database[7]. WordNet organizes the English nouns, verbs and adjectives into synonym sets (synsets), which represent a unique lexical concept. A given English word can belong to multiple synsets and conversely, each synset has multiple words or word forms, which are synonyms of each other. WordNet supports different relationships between synsets like hypernym/hyponym, synonym/antonym, meronym/holonym etc. In our prior work [12], we exploited the hypernym/hyponym relationship between synsets to compute the implication distance measure between two synsets.

In this work, semantics are incorporated through the use of ConceptNet. ConceptNet is a large repository of common-sense knowledge. ConceptNet supports 20 different semantic relationships between concepts like “capableOf”, “effectOf”, “user-For”, “isA”, “partOf” etc. We use ConceptNet, as it captures a very rich set of semantic relationships as compared to WordNet. This leads to a very uniform distribution of implication distance measures between two synsets, when computed using Conceptnet instead of WordNet [10].

Semantic Distance. In this subsection, we shall describe the procedure to compute the implication distance measure between two synsets s_i and s_j using ConceptNet. A WordNet synset has multiple synonym words associated with it. Let us assume that synset s_i is associated with words $w_{i,1}$, $w_{i,2}$ and $w_{i,3}$; and synset s_j is associated with words $w_{j,1}$, $w_{j,2}$ and $w_{j,3}$. Let us also assume that $d_c(w_1, w_2)$ denotes the ConceptNet distance between two words (concepts) [1]. The distance between two synsets s_i and s_j is then given as:

$$d(s_i, s_j) = \max_q \min_k d_c(w_{i,q}, w_{j,k}), \quad (2)$$

where $w_{i,q}$ is the word associated with synset s_i and $w_{j,k}$ is the word associated with synset s_j . The distance $d(s_i, s_j)$ is the hausdorff distance between two sets containing synonym words. The implication measure between two synsets s_i and s_j is given as:

$$I(s_i, s_j) = 1 - d(s_i, s_j), \quad (3)$$

where $d(s_i, s_j)$ is as defined in equation (2).

3 Propagation of Semantics

In this section, we present the details of our algorithm on semantic propagation. Let us assume that the user wishes to annotate an image a and that there are N images in the database. When the user enters annotations for image a , she is asked to fix the sense (i.e. the semantics) of the word that she is using for annotation using WordNet. For example, if she annotates an image with the word “suit”, then she fixes the sense to be either a “lawsuit” or “clothing” or “pack of cards” etc. Fixing the sense of the word exploits the hierarchical relationship among synsets in WordNet. The current image a is considered as a positive example image for this synset.

3.1 Algorithm Details

We shall now discuss three key aspects of the algorithm – (a) local trees, which identify the semantic space of the fixed synset and help in computing semantic distance between synsets, (b) propagation of the synsets across the database and (c) providing feedback by presenting the user with an image that will maximize the rate of semantic propagation.

Local Trees. For each fixed synset k , we need to define a local tree T_k . A local tree is a subset of the WordNet ontology. It is a hierarchy of nodes, with synset k being the node at the center of the hierarchy. Synset k is also called the root node of the tree. Nodes above the center are the generalizations of synset k and nodes below it are its specializations. Formally, a local tree T_k is defined as follows:

$$T_k = \{s \mid d(s, k) \leq m\}, \quad (4)$$

where $d(s, k)$ is the hop distance between the node representing synset s and synset k , and m is the number of specialization and generalization levels to be considered. In our case, we set $m = 2$, based on a trade off between computational complexity and accuracy.

Thus a local tree efficiently partitions the semantic space of WordNet – it helps to quickly identify if two synsets are semantically far apart.

Computing Semantic Propagation Likelihoods. The new synset k , entered by the user is then propagated to other images based on low-level features using color, texture and edge histograms, WordNet local trees and ConceptNet implication measure [ref. Section 2.2]. This is done as follows: we determine $L_f(kli)$, that is the low-level feature likelihood that image i belongs to synset k .

$$L_f(k|i) = L_f(k^+|i) - L_f(k^-|i), \quad (5)$$

where, k^+ denotes the set of all the positive example images directly associated with synset k , as well as the positive example images associated with the synsets present in the local tree of k . k^- denotes the set of all the negative example images of synset k , as well as all the positive example images of the synsets, not present in the local tree of k . $L_f(k^+|i)$ denotes the low-level likelihood that the image i belongs to the set k^+ . $L_f(k^-|i)$ denotes the low-level likelihood that the image i belongs to the set k^- . $L_f(k^+|i)$ is defined as follows:

$$L_f(k^+|i) = \sum_{j=1}^P \exp(-\beta d_{ij}) I(k, s_j), \quad s_j \in T_k, \quad (6)$$

where P is the total number of synsets in T_k (local tree of synset k) [ref. Section 3.1], whose positive examples are being considered. d_{ij} is the average low-level feature distance between image i and the positive examples of synset s_j . β is a constant and $I(k, s_j)$ is the implication between synsets k and s_j using ConceptNet [ref Section 2.2]. $L_f(k^-|i)$, denotes the low-level likelihood that the image i belongs to set k^- and is given as follows:

$$L_f(k^-|i) = w_1 \exp(-\beta d_{ik}) + \frac{w_2}{N} \sum_{j=1}^Q L_f(s_j^+|i), \quad s_j \notin T_k, \quad (7)$$

where β is a constant, d_{ik} is the average low-level feature distance between image i and the negative examples of synset k . w_1 and w_2 are weights where $w_1 + w_2 = 1$ and Q is the total number of synsets not in the local tree of synset k .

We now show how to compute the ConceptNet likelihood. This is done by calculating the likelihood of other synsets already present in the database (but not manually associated with the image) to image i . These other synsets are present in the database, as the user has entered them as manual annotations for some other images in the database. This likelihood is the ConceptNet likelihood and is given as follows:

$$C_l(s_1, s_2, \dots, s_M | k) = \frac{1}{M} \sum_{j=1}^M I(s_j, k), \quad (8)$$

$$L_c(k|i) = C_l(s_1, s_2, \dots, s_M | k)$$

where s_1, s_2, \dots, s_M are the synsets which are manually associated with the image i by the user. $I(s_j, k)$ is the ConceptNet implication between manual synset s_j and synset k which we want to propagate and M is the total number of synsets manually associated with image i .

Feedback. The system now presents an image to the user for relevance feedback. This is an image that is least likely to be associated with the annotations that accurately reflect the semantics of the image. The final likelihood, for picking the least likely image, for an image i is computed as follows:

$$l_i = \frac{1}{M} \sum_{j=1}^M \alpha L_f(s_j|i) + \beta L_c(s_j|i), \quad (9)$$

where M is the number of synsets which were automatically propagated to image i . $L_l(s_j|i)$ is the low-level feature likelihood of image i with respect to synset s_j , and $L_c(s_j|i)$ is the ConceptNet likelihood of image i with respect to synset s_j . α and β are constants where $\alpha+\beta=1$. The least likely image j^* is picked as follows:

$$j^* = \arg \min_j l_j, \tag{10}$$

where j varies over the total number of images in the database, and l_j is the final likelihood of image j .

During relevance feedback, the user can either delete an associated synset, or confirm an automatically propagated synset or add another synset. If the user deletes an associated synset, then we mark the image as a negative example of the synset. We then recompute the distance of every other image in the database, with respect to the newly updated positive and negative example sets of that synset. On the other hand, if the user confirms an automatically associated synset, then we consider this image as a positive example of the synset, and follow the same procedure of recomputing the distance of every other image, as explained earlier. Same holds true if the user adds a synset to the image. Thus with each iteration of relevance feedback the association between images and synsets gets refined, and becomes more accurate with respect to the image semantics.

In this section, we have discussed our annotation algorithm in detail. The algorithm focuses on (a) semantic propagation using ConceptNet and WordNet and (b) maximizing the rate of propagation by providing relevance feedback for the least likely image. We now describe a front end that uses the algorithm as the computational backbone and allows the user to annotate media.

4 Visual Annotation Interface

In this section we shall discuss the visualization interface that allows users to group images based on concepts without manual annotations. Here a *concept* is an abstract idea that the user associates with an image. For example the user can associate a concept of “peace” with the image of a dove.

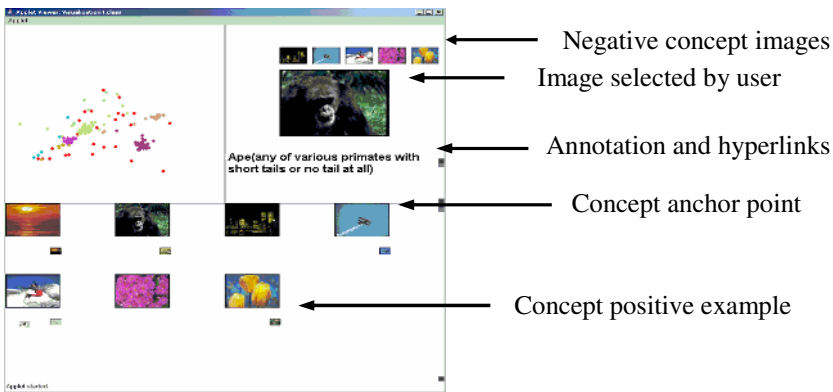


Fig. 1. Visual Annotation Interface

Our user interface is shown in Figure.1 The left panel of the interface shows the images in the form of small circles, colored with the dominant (mean) color of the image. The bottom pane shows the various concepts created in the system and their positive examples. As the user moves the mouse over the image circles in the left pane, the right pane shows the details viz. the image, the concept images of which the image is a positive or a negative example, and the annotations of the image. Note that this interface is in its preliminary stage and we plan to conduct extensive evaluation of the various aspects of the interface in the future.

4.1 Creating Visual Concepts

Initially all the images in the form of circles are scattered at random in the interface. Now the user can click on one of these image circles and create a concept in the system. The clicked image is considered to be a positive example of the concept and is treated as an anchor point for the concept. Every concept is represented by a unique color in the system. All image circles representing images belonging to the same concept are colored with the same color to indicate semantic closeness.

Visual Feedback. After the creation of the first concept, all the other unlabeled images in the database viz. images that are neither positive nor negative examples of any concept, move towards the anchor point of this concept. This movement is based on their color histogram distance with respect to the positive (and negative) examples of the concept. However, if the color histogram distance is above a threshold; the image movement is restricted to be δ of the current display distance between the image and the anchor point. This is done to avoid cluttering and to prevent all images getting merged into a single image circle on screen. Thus a concept cluster is formed with the anchor point being the center of the cluster.

Creating Positive Visual Clusters. The user can then add positive examples to the concept by dragging the image circles on the concept anchor point in the bottom pane. If the color histogram distance between the dragged image and the positive example images of the concept is within a certain threshold, the image circle will move towards an already established concept cluster, else it will create another cluster for the same concept by being another anchor point. Thus you can have multiple clusters of the same concept which are visually at a larger display distance but semantically close. For example, if the user first created a concept anchor point with an image of a red flower, and then associated the image of a yellow flower with it then they would visually appear at different points in the pane, but the color of the image circles would still be same (since they belong to the same *concept*). Also, all the unlabeled images will move towards the closest cluster of the closest concept by the same mechanism as above. The closest concept c^* is then defined as the one that maximizes the difference between positive and negative likelihood for the unlabelled image.

$$c^* = \arg \max_c diff_c^*, \quad (11)$$

$$diff_c = L_f(c^+ | i)(1 - L_f(c^- | i)),$$

where c^+ denotes the set of positive examples associated with visual concept c and c^- denotes the set of negative examples associated with concept c . $L_f(c^+ | i)$ is the positive

feature likelihood of the unlabelled image i with respect to the set c^+ and $L_f(c^-|i)$ is the negative feature likelihood of the unlabelled image i with respect to the set c^- . $L_f(c^+|i)$ is then given as:

$$L_f(c^+|i) = \frac{1}{M} \sum_{j=1}^M \exp(-\beta d_{ij}), \quad (12)$$

where β is a constant, M is the total number of positive examples of the visual concept c , and d_{ij} is the color histogram distance between images i and j . $L_f(c^-|i)$ is given as:

$$L_f(c^-|i) = \frac{1}{N} \sum_{k=1}^N \exp(-\beta d_{ik}), \quad (13)$$

where β is a constant, N is the total number of negative examples of the visual concept c and d_{ik} is the color histogram distance between images i and k . The closest cluster within the closest concept c^* is then determined as the one that minimizes the feature distance between the image i and the positive examples of the cluster.

Creating Negative Visual Clusters. The user can also specify negative examples for a visual concept. This creates a negative cluster for the concept and the image becomes a negative anchor point or moves towards the anchor point of an already created negative cluster of the concept. If the image becomes a negative anchor point then it moves away from the closest cluster of the concept by an amount equal to the average feature distance between the image and the positive examples of the cluster. If the average feature distance is above a certain threshold then the moving away is restricted to be δ of the current display distance so that the image circles don't go out of the display screen. Unlike multiple positive clusters per concept, there is only one negative cluster per concept. When an image becomes a negative example, it is duplicated on screen and is colored in red. So, all negative clusters of all visual concepts appear in red. Also, an image could be a positive or a negative example of more than one visual concept, in which case it is duplicated on screen.

4.2 Associating Annotations to Visual Concepts

The user can also add annotations to visual concepts by clicking on the concept anchor point in the bottom pane. As the user enters the annotations, she is asked to pick the sense of the word using WordNet. These senses (synsets) are then propagated to other unannotated images in the system based on feature likelihood and WordNet likelihood. Let us assume that the user annotates the visual concept c with the synset k . The positive examples of the concept c then become the positive examples of synset k and negative examples of c become negative examples of k . The system then propagates synset k to all the unannotated images that had moved towards concept c , with a feature likelihood, $L_f(k|i)$ that is given as:

$$L_f(k|i) = L_f(c^+|i) - L_f(c^-|i), \quad (14)$$

where $L_f(c^+|i)$ is the positive feature likelihood of image i with respect to the set c^+ and is given as:

$$L_f(c^+ | i) = \exp(-\beta \frac{1}{M} \sum_{j=1}^M d_{ij}), \quad (15)$$

where β is a constant and M is the number of positive examples associated with concept c and d_{ij} is the feature distance between images i and j . $L_f(c^- | i)$ is the negative feature likelihood of image i with respect to set c^- and is given as:

$$L_f(c^- | i) = \exp(-\beta \frac{1}{N} \sum_{j=1}^N d_{ik}), \quad (16)$$

where N is the total number of negative examples associated with concept c .

Propagating Synset to Other Visual Concepts. The system then propagates synset k to all the other images in the system that are neither positive, negative nor automatic examples of visual concept c . For every such image i , the system determines all the distinct visual concepts to which image i belongs. Let us denote one such visual concept as v . The system then determines the feature likelihood of synset k with respect to the visual concept v , $L_f(k | v)$ as:

$$L_f(k | v) = L_f(k^+ | v) - L_f(k^- | v), \quad (17)$$

where k^+ denotes the positive examples associated with synset k , and k^- denotes the negative examples of synset k . $L_f(k^+ | v)$ denotes the feature likelihood of the visual concept v with respect to the set k^+ and is given as:

$$L_f(k^+ | v) = \sum_{j=1}^M \exp(-\beta d_{vj}) I(k, s_j), \quad s_j \in T_k, \quad (18)$$

where β is a constant and M is the total number of synsets in the local tree of k , i.e. T_k [ref. Section 3.1] whose positive examples are being considered and d_{vj} is the average feature distance between all the positive examples of visual concept v and positive examples of synset s_j . Similarly, the feature likelihood of the visual concept v with respect to the set k^- is given as:

$$L_f(k^- | v) = w_1 \exp(-\beta d_{vk}) + \frac{w_2}{N} \sum_{j=1}^N L_f(k^+ | v), \quad s_j \notin T_k, \quad (19)$$

where β is a constant and d_{vk} is the average feature distance between positive examples of visual concept v and the negative examples of synset k and N is the total number of synsets not in the local tree of synset k . The feature likelihood of synset k with respect to image i , $L_f(k | i)$ is then given as:

$$L_f(k | i) = \frac{1}{M} \sum_{j=1}^M L_f(v_j | i) L_f(k | v_j), \quad (20)$$

where $L_f(v_j | i)$ is the feature likelihood that image i belongs to visual concept v and M is the total number of visual concepts to which image i belongs. The system then calculates the WordNet likelihood of other synsets already present in the database, but

not manually associated with the positive examples of the visual concept c . These other synsets are present in the database, as the user has entered them as manual annotations for some other visual concepts in the database. This wordnet likelihood for an image i which is a positive example of concept c , is then given as:

$$W_i(s_1, s_2, \dots, s_M | k) = \frac{1}{M} \sum_{j=1}^M I(s_j, k), \quad (21)$$

$$L_w(k | i) = W_i(s_1, s_2, \dots, s_M | k)$$

where s_1, s_2, \dots, s_M are synsets which are manually associated with image i . $I(s_j, k)$ is the Wordnet implication [12] between manual synset s_j and synset k which we want to propagate and M is the total number of synsets manually associated with image i .

In this section, we have described an interface that allows the user to create concepts, without entering annotations. It does not require the presence of high quality text (annotations) to organize images. Moreover, the user is only required to drag-and-drop in order to expand the positive and the negative image data set of a concept and thus organize images. This is much more intuitive and easier than entering text for every image. We shall now discuss our experimental results and evaluation scheme.

5 Experiments

The experiments were conducted on three different datasets – (a) a set of 242 images containing 90 distinct ground truth synsets, (b) a set of 500 personal image collection containing 107 distinct ground truth synsets and (c) a set of 1000 Corel image collection consisting of 44 distinct ground truth synsets. The ground truth for these image sets was created manually, by fixing the sense of the annotation. We refer to this fixed sense of the ground truth annotation as the ground truth synset. The entire prototype was developed in Visual J# in Microsoft Windows platform.

We now enumerate the steps taken to test our system:

- In order to test the system, we created an automatic test script that simulated a user annotating the images and providing relevance feedback. We picked a random image initially.
- We exposed the ground truth of this random image and annotated it with the ground truth synset. This is equivalent to the user picking an image and annotating it.
- After performing all the required propagations [ref. Section 3.1, equation(5),(8)], the system picked an image that was least likely to be associated accurately, with the semantics of its annotations, and uncovered its ground truth.
- If the automatic annotation synsets of this image coincide with the ground truth, then the system confirms the automatic annotations and updates the positive example images for that synset.
- However, if the automatic synsets and the ground truth synsets don't match, i.e. they are not present in each other's local tree [ref. Section 3.1], then the system considers the image as a negative example of the automatically

propagated synsets, and a positive example of the ground truth synset. This is equivalent to the user giving relevance feedback for the least likely image, where he deletes the automatically propagated synsets, and adds the ground truth synsets to the image.

- Also, if the ground truth is present in the local tree of the automatically propagated synset, then the system treats the image as a weaker positive example of the automatically propagated synset and updates its likelihood with respect to the automatic synsets. This is intuitive because of the semantic relationship between words that is captured by WordNet ontology. For example, if the ground truth of an image is “car” and the automatic propagation is “automobile” then since “car” is present in the local tree of “automobile”, as “car” is semantically related to “automobile” through is-a relationship, we can treat the image as a weaker positive example of “automobile”.

5.1 Evaluation

We chose to test our system by defining a new evaluation scheme. We used Average ConceptNet similarity measure as opposed to precision-recall. This is done since precision-recall does not take into account the semantic relationship between concepts and is therefore inadequate in determining the performance of our algorithm. This is intuitive since an image of “flower”, mislabeled as “plant” is not a large error and this fact should be addressed by the evaluation procedure.

We now describe our evaluation mechanism. During each iteration, the ConceptNet similarity between automatically propagated synsets and the ground truth was computed for all images. Let us assume that image i has N automatically propagated synsets. The system then picked a subset G , of these automatic synsets such that

$$G = \{k \mid L_f(k \mid i) \geq \alpha\}, \quad (22)$$

where α is a constant and $L_f(k \mid i)$ is the feature likelihood of propagating synset k to image i . This is intuitive since we do not want to consider those synsets which have a very low propagation likelihood, as they do not help in bridging the gap between semantics and low-level features of an image and in tasks such as search. The system then computed the expected ConceptNet similarity s_i for an image i as:

$$\begin{aligned} s_i &= 1 - \min_{k,j} d(g_{i,j}, a_{i,k}), \quad k \in G, G \neq \emptyset, j \in 1..M, \\ s_i &= 0, \quad \text{if } G = \emptyset, \end{aligned} \quad (23)$$

where M is the number of ground truth synsets associated with image i , $a_{i,k}$ is the automatically propagated synset, $g_{i,j}$ is the ground truth synset and $d(g_{i,j}, a_{i,k})$ is the distance measure between two synsets computed using ConceptNet [ref. Section 2.2].

The ConceptNet similarity was then averaged over all the images in the database to determine the performance of the algorithm. The average ConceptNet similarity is:

$$\overline{D_c} = \frac{1}{M} \sum_{i=1}^M s_i, \quad (24)$$

where M is the total number of images in the database. We plotted the average ConceptNet similarity against number of iterations for three different data sets when only

one least likely image was picked for relevance feedback. This was done to mimic an ordinary end user who provides feedback on only one image.

Fig. 2 show the ConceptNet similarity graphs for an optimized value of α that was determined by extensively testing the system for different values of α . The graphs represent an average of 10 experiments, each carried out with a different initial random image. This was done to avoid the chance exposure of an initial image that could be semantically close to the other images in the database.

The baseline similarity curve in the graphs indicates the lower bound on the performance of the algorithm. The baseline similarity was computed by exposing and annotating a single image in each relevance feedback cycle without any semantic propagation. So, after exposing k images, the baseline similarity will be k/M , where M is the total number of images in the database.

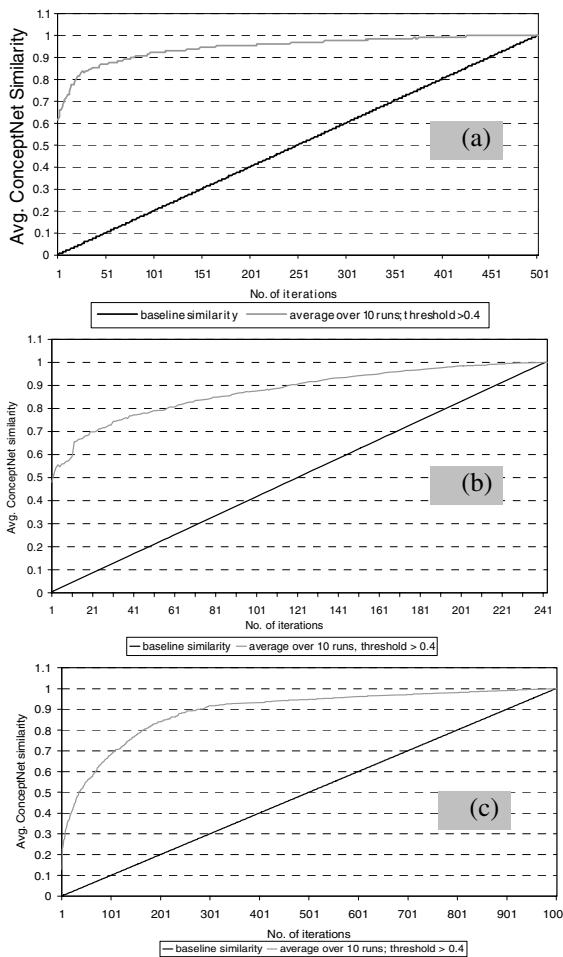


Fig. 2. Average ConceptNet similarity against No. of iterations for – (a) 242, (b) 500 and (c) 1000 images. Ground truth contains 90, 107 and 44 distinct WordNet synsets respectively.

As the graphs suggest, the semantic similarity between the annotations and the images increases with an increase in relevance feedback iterations. This is intuitive since we take into account the semantic relationships between annotations, using ConceptNet similarity measures that work very well, as ConceptNet captures a wide variety of semantic relationships.

Our results compare well with related work [2,14]. For example, in [14] the authors show that the algorithm converges to 90% accuracy within four iterations. However, in *each* iteration, the system evaluates a 100 images for relevance feedback. In our system, we achieve a ConceptNet similarity of 0.8, using only 58/242 iterations. In [2] the authors report achieving a 50% accuracy by annotating only 20% of the images; this compares well with our result. However, since the authors use only classmembership and not ConceptNet to present their results, we believe that their results will improve with the use of ConceptNet.

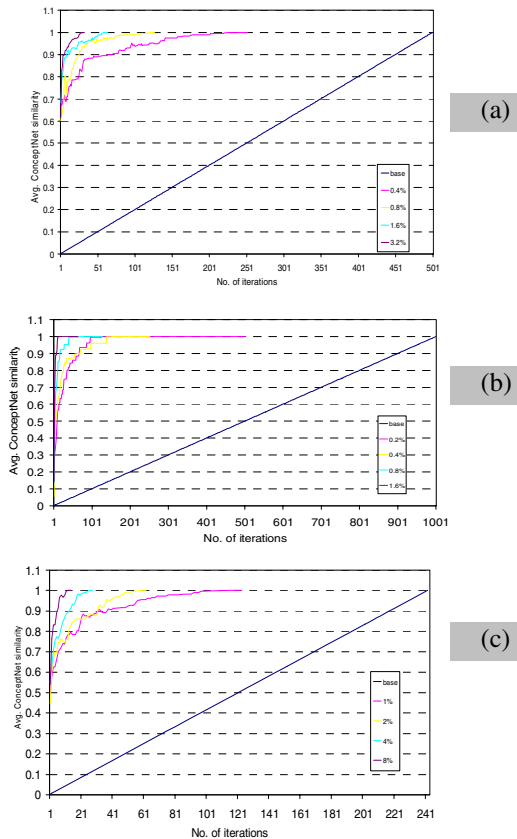


Fig. 3. Average ConceptNet similarity v/s No. of iterations, when different percentages of images are exposed as least likely for relevance feedback on the dataset of – (a) 242 images, (b) 500 personal image collection and (c) Corel dataset of 1000 images

Providing Relevance Feedback for Larger Number of Images. We also conducted experiments to study the performance of the algorithm when different fraction of the images were exposed in each cycle for relevance feedback. Figure 3 shows the average ConceptNet similarity graphs for the three datasets when different percentage of images were picked as least likely.

As the graph suggests, the semantic similarity rises faster with an increase in number of images that are exposed for relevance feedback. This is intuitive since higher number of synsets get introduced and propagated in the system at each relevance feedback cycle and therefore, the similarity between the annotations and the image semantics rises faster.

6 Conclusion

In this paper, we have presented – (a) a novel strategy for semantic propagation, (b) a visual annotation interface and (c) novel evaluation scheme. The semantic propagation is done using – (1) low-level features (2) WordNet ontology, (3) ConceptNet repository and (4) relevance feedback. The visualization interface allows the user to browse, organize and annotate a large collection of multimedia images using *visual concepts*. This interface is interactive, real-time and scalable. Our evaluation scheme was based on ConceptNet similarity measure and *not* on precision-recall as we believe that precision-recall does not fully utilize the linguistic relations amongst words when evaluating the performance of our annotation algorithm.

We then conducted experiments to show how the semantics propagate across the database. The results indicate that the system performs much better than the baseline case, and as the number of relevance feedback cycles increases, the semantic association between the images and the annotations, becomes more refined and accurate.

In the future, we plan on incorporating sophisticated machine learning algorithms that use Support Vector Machines, for better semantic propagation. We are also looking at incorporating user-context for personal ontologies and incorporating event structures associated with personal media collection into the system. We also plan to conduct extensive evaluation of our visual annotation interface.

References

1. P. Appan, B. Shevade, H. Sundaram and D. Birchfield (2005). *Interfaces for networked media exploration and collaborative annotation*, Proc. Int. Conf. on Intelligent User interfaces, also AME-TR-2004-11, Jan. 2005, San Diego, CA.
2. E. Chang, K. Goh, G. Sychay and G. Wu (2003). *CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines*. IEEE Transactions on Circuits and Systems for Video Technology **13**(1): 26-38.
3. X. He, W.-Y. Ma, O. King, M. Li and H. Zhang (2002). *Learning and inferring a semantic space from user's relevance feedback for image retrieval*, Proc. of the 10th international conference on Multimedia, 343-346, Dec. 2002, Juan Les-Pins, France.
4. A. K. Jain (1989). *Fundamentals of digital image processing*. Prentice Hall Englewood Cliffs, NJ.

5. J. Li, C. Plaisant and B. Shneiderman (1998). *Data Object and Label Placement for Information Abundant Visualizations.*, Workshop on New Paradigms in Information Visualization and Manipulation (NPIV '98), ACM, 41-48, New York.
6. H. Liu and P. Singh (2004). *ConceptNet: a practical commonsense reasoning toolkit.* BT Technology Journal **22**(4): pp. 211-226.
7. G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller (1993). *Introduction to WordNet: An On-line Lexical Database.* International Journal of Lexicography **3**(4): 235-244.
8. Y. Rui and T. Huang (1999). *A Novel Relevance Feedback Technique in Image Retrieval.*, Proc. ACM Multimedia 1999, Nov. 1999, Orlando, FL.
9. B. Shevade and H. Sundaram (2003). *Vidya: An Experiential Annotation System.* 1st ACM Workshop on Experiential Telepresence, in conjunction with ACM Multimedia 2003, AME-TR-2003-04, Nov. 2003., Berkeley, CA.
10. B. Shevade and H. Sundaram (2005). *A Visual Annotation Framework using Common-Sensical and Linguistic Relationship for Semantic Media Retrieval.* AME-TR-07,2005.
11. B. Shneiderman and H. Kang (2000). *Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos.* In: Proc. International Conference Information Visualization (IV2000). London, England,
12. H. Sridharan, H. Sundaram and T. Rikakis (2003). *Context, memory and Hyper-mediation in Experiential Systems.* 1st ACM Workshop on Experiential Telepresence, in conjunction with ACM Multimedia 2003., AME-TR-2003-02, Nov. 2003., Berkeley CA.
13. L. Wenyin, S. Dumais, Y. Sen, H. Zhang, M. Czerwinski, et al. (2001). *Semi-Automatic Image Annotation.* Proc. Human-Computer Interaction--Interact 01, pp.326-333,
14. L. Ye, C. Hu, X. Zhu, H. Zhang and Q. Yang (2000). *A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems.*, In: Proc. ACM MM2000, 31-38, Nov. 2000, Los Angeles, CA.

Improving Access to Multimedia Using Multi-source Hierarchical Meta-data

Trevor P. Martin^{1,2} and Yun Shen¹

¹Artificial Intelligence Group, Dept of Engineering Maths,
University of Bristol, UK
{trevor.martin, yun.shen}@bris.ac.uk

²Currently Senior Research Fellow, Computational Intelligence Group,
Intelligent Systems Lab, BT Research and Venturing, Adastral Park, Ipswich, UK

Abstract. Efficient retrieval of multi-media content depends on the availability of adequate meta-data to indicate the nature of the content. Such meta-data often contains useful hierarchical categorisation but is frequently not consistent between different sources. We outline a method to identify equivalent instances which are described by different meta-data schemata, and to find correspondences between hierarchies, which may be used to improve instance matching. Some successful initial tests of the method on large movie databases are reported.

1 Introduction

Efficient retrieval of multi-media content depends on the availability of adequate meta-data to indicate the nature of the content. Such meta-data may be created in advance of use or computed dynamically from the content (this is a broad view of meta-data which includes any features or summaries derived from, or describing, the content). However, there are few standards for meta-data of this sort, and there may be even less agreement on how the meta-data should describe the content. Even a straightforward schema such as the Dublin Core allows for free text in parts of the description, and hence there is a degree of subjectivity. Although this can be restricted to some degree by use of controlled vocabularies, free text almost inevitably leads to a problem in matching user queries with a data source, or in combining data from more than one source. Clearly this problem is not unique to multi-media retrieval, and the work reported here is widely applicable.

It is clear that in natural language, the association between symbols and objects is not necessarily unique or one-to-one. For example, “Elton John”, “Reg Dwight” and “the writer of the music of ‘Candle in the Wind’” all denote the same person and could be used interchangeably as meta-data, i.e. they are semantically identical even though they are not syntactically identical. The word “Europe” denotes a collection of countries, but its precise definition is elusive - is it a set of countries marked as Europe on a particular map, the members of the European union (now? in 1970 ? in 1975 ? in 2010 ?), the countries eligible to enter European Championship football, countries eligible to enter the Eurovision song contest ? In this case, semantically distinct entities are denoted by syntactically identical words.

Such questions may be dismissed as belonging to philosophy and linguistics, but have direct relevance to computerised information systems and the flexible management of information. Most information systems are built (explicitly or implicitly) using crisply defined categories and rely on the underlying theory of databases, which requires a unique identifier for every individual entity.

Within a single database or information source, this can work adequately (although there are still problems if an object is assigned more than one “unique” identifier). The problem becomes more serious when attempting to combine information from multiple sources - we face the problem of not knowing whether the different sources are referring to the same object or not. Additionally data may be in slightly different formats, increasing the problem of matching.

The classification structure and attributes (properties) of the objects (i.e. the values associated with meta-data tags) can be used to guide searching and integration of multiple sources. Even if different hierarchies use different categories, there is likely to be a degree of correspondence, and objects placed within similar categories are likely to have similar properties. For example, a digital library and an online book-seller refer to the same (structured) objects but may differ in categorisation and details stored about each book.

The properties can be used to group the objects into classes, which may in turn form some sort of hierarchical structure. This leads to the “ontology alignment” or “schema matching” problem; the method outlined here uses “instance-matching” initially and is then extended to use and/or predict the hierarchical classification.

In this paper, we have tested the instance-matching on a database of nes stories; we then consider two meta-data sources which describe films using different sets of tags and (importantly) different genre hierarchies to classify the films. Given mappings between meta-data attributes, we

- (i) outline a way of automatically identifying equivalent instances
- (ii) use this to learn a soft mapping between genre hierarchies
- (iii) use the genre map to improve detection of equivalent instances

2 Background – Record Linkage and Schema Matching

The problem of record linkage was identified in the USA public health area, when combining different records that (possibly) referred to the same patient. Newcombe [17] proposed a frequency-based approach which was later formalised by Fellegi and Sunter [11]. These approaches assume that the two data sources have common attributes, and are commonly applied to the so-called “merge/purge” problem in business databases to filter out duplicate entries. The methods focus on calculating a weight for each attribute in the database, according to the likelihood of finding matching values within that attribute’s domain (i.e. the set of all values appearing in the column).

The initial formulation treated binary matches (true/false) but was extended to categorical matches (one of a small set of values) and continuous matches (e.g. a number in the interval $[0, 1]$). By assuming conditional independence between records matching on different attributes it is possible to estimate the conditional probabilities for each attribute matching, given that the records are (or are not) identical, and hence to find thresholds for classifying two records as matching or not according to the

weighted sum of matches. The estimation can be on the basis of minimum error probabilities, expectation maximisation, utility (cost of incorrect decision) etc – see [10] for an overview.

These methods implicitly take into account knowledge of the database schema, as they assume each record consists of the same set of attributes.

The record linkage problem was extended to analytic linkage (also referred to as entity matching) by considering the combination of data taken from two or more sources e.g. the integration of heterogeneous databases. Dey et al [7] give a summary of probabilistic approaches, based on the same framework as the record linkage work outlined in the previous paragraph. Again, knowledge of the schema is assumed in that matching pairs of attributes are known.

These methods use several techniques to try to match attributes, such as standardising the form of names and addresses, and applying heuristics (for example first-n-characters match, common substrings, edit distance is below a specified threshold). Bilenko, Mooney et al [4] describe SoftTF-IDF, an adaptive matching function, which takes account of the frequencies of similar and identical words within a domain.

The problem can also be approached at the schema level, by looking at labels (i.e. attribute names) and constraints associated with allowed values.

Rahm and Bernstein [18] provide a useful survey of approaches to the automation of schema matching, including both database and semantic web problems. They identify three main groups, with methods arising from the fields of:

- information retrieval – using distance-based matching techniques such as the edit distance to overcome the inadequacy of exact, “keyword-based” matching. These assume the use of fairly simple mappings between attribute domains.
- machine learning – using algorithms to create a mapping between attributes based on the similarity among their associated values. Bayesian classifiers are the most common approaches (e.g., GLUE [9] and Autoplex [3])
- graph theory – by representing schemata in tree or graph form, e.g. the TreeMatch algorithm [14] which estimates the similarity of leaf nodes in an XML DTD by estimating the similarity of their ancestors.

They also present a taxonomy covering many existing approaches based on the split between metadata matching and content (instance) matching as above, but also explicitly considering other aspects of matching such as

- structure matching (e.g. an attribute such as address in one source could map to several attributes such as street, town and postcode in another)
- hierarchical and similarity-based matching of terms (e.g. book is a type of publication, make and brand are roughly synonymous)
- matching based on data constraints (e.g. DoB is of type Date and so could match an attribute BirthDate in another source, which is also of type Date).

We argue that schema and instance matching systems need to include some measure of uncertainty in the matching process, such as Cupid’s “plausibility factor” [14]. The need for uncertainty has also been noted by others. For example, Chang and

García-Molina [6] developed an approach for the precise translation of Boolean queries across different information sources. In a subsequent paper [5] they presented a real-world case study (combining book searches from four different web sites), and found that it was only possible to make exact mappings in 30% of the rules - the remaining 70% required approximation.

Gal et al [12] recognised the need to include uncertainty in the matching process, and outlined a fuzzy framework for schema integration based on the notion of similarity relations. Gal has also looked at the problem of evaluating the matching between schemata, compared to a notional “ideal” matching that would be produced by a human.

Ding and Foo [8] reviewed the problem from a semantic web perspective, noting that “current researches on semi-automatic or automatic ontology research in all three aspects of generation, mapping and evolving have so far achieved limited success.”

Madhavan, Bernstein et al [13] outlined a way of defining languages to represent mappings between schemas and ontologies (domain models in their paper), and concluded by saying that

“the next step in this work is to develop an appropriate probabilistic representation of mappings that enables capturing both inaccuracy and uncertainty”

The integration of semi-structured information from heterogeneous sources is an unsolved and important problem which is ideally suited to the techniques of soft computing (see also [15]). As an illustration, we consider below the problem of finding “identical” movies from descriptive meta-data derived from two sources. We first outline the method used, and then show its application to the movie meta-data.

3 SOFT – A Structured Object Fusion Toolkit

The SOFT approach is described in detail in [16], we give a brief overview here. Assume we have two sets of objects $A = \{a_1 \dots a_n\}$ and $B = \{b_1 \dots b_m\}$, and that we wish to establish an approximate relation

$$h : A \rightarrow B \quad (1)$$

We envisage an underlying universe U (the “real world”) which is modelled by the objects in sets A and B , i.e. there are functions

$$f : U \rightarrow A \quad (2)$$

and

$$g : U \rightarrow B \quad (3)$$

which are unknown and in practice may be incomplete and/or “noisy” in some way, meaning that some or all of the mappings f and g may be incorrect. We seek the best approximation to h such that

$$h(f(x)) = g(x) \quad (4)$$

i.e. we seek to establish a correspondence between the objects in A and B , based on the properties (attributes) of the objects.

Let the objects in A and B have attribute values taken from $C_1, C_2, \dots, D_1, D_2, \dots$ with relations defined as

$$R_i : A \rightarrow C_i \quad i=1 \dots n_A$$

$$S_j : B \rightarrow D_j \quad j=1 \dots n_B$$

Note that these are relations, i.e. they can be single- or multi-valued. Examples would be *title, director, year* etc.

We do not assume that the information about A and B in relations R_i, S_j is identical or completely consistent, but we do assume that some of these relations reflect similar or identical properties of the objects in A and B . Thus for some choices of pairs of codomains (C_i, D_j) we assume an exact or approximate matching function h_{ij} which for each element of C_i returns a (possibly fuzzy) subset of D_j . As shown in [1, 2], this can be converted to a mass assignment giving an estimate of the probability that the element corresponding to some C_i lies in a subset $\{d_1 \dots d_k\} \subseteq D_j$. (We will refer to h_{ij} as a function even though its output is not necessarily a single value.)

The h_{ij} can be obvious mappings from codomain to codomain, involving exact matches or small permutations, truncations, etc; alternatively they can be more sophisticated functions, possibly the product of a machine learning process. The proportion of a domain that matches under such a mapping gives an indication of the “overlap” between domains and hence the possibility that two attributes correspond.

This is obvious if the functions h_{ij} are exactly known – for each object a_k in A ,

$$h(a_k) = S_j^{-1}(h_{ij}(R_i(a_k))) \tag{5}$$

i.e. we obtain a subset of objects in B which possibly correspond to the object a_k .

If we can establish a reasonably specific matching h_{ij} between two codomains C_i and D_j , we can use this to refine the current approximation to h , since if

$$R_i(a_k) = C_{ik}$$

and $h_{ij}(C_{ik}) = \tilde{D}_{jk}$ where \tilde{D}_{jk} denotes a fuzzy subset of D_j

and $S_j(\tilde{B}_k) = \tilde{D}_{jk}$ using the (inverse) extension principle

then $h(a_k) = \tilde{B}_k$

i.e. a_k corresponds to a fuzzy subset \tilde{B}_k . We consider each h_{ij} to give a different “observation” (or sample) of the true value¹, and seek the fuzzy set which is most likely to give these observations. In order to achieve this, we use mass assignment theory [1, 2].

Consider a number of observations which are converted from fuzzy sets to mass assignments over the universe of possible values B .

Let M_n be the mass assignment on B that makes the observed values most likely after n observations, i.e. choose the masses to maximize

¹ In an analogous way, to determine the bias of an unfair coin we could observe the results of several sequences of coin tosses and choose a bias to maximise the likelihood of the observations.

$$Pr(M_n|o_1, o_2, \dots, o_n)$$

This gives a way of updating M after each observation. Using a naive bayes assumption²

$$Pr(M_n|o_1, o_2, \dots, o_n) = \frac{Pr(o_1, o_2, \dots, o_n|M_n) \times Pr(M_n)}{Pr(o_1, o_2, \dots, o_n)} \tag{6}$$

$$Pr(o_1, o_2, \dots, o_n|M_n) = Pr(o_1|M_n) \times Pr(o_2|M_n) \times \dots \times Pr(o_n|M_n)$$

Assuming each possible mass assignment M_n is equally likely,

$$M_n(B_k) = \frac{N_n(B_k)}{\sum_{X \subseteq B} N_n(X)} \tag{7}$$

where $N_n(X)$ is number of times the subset X has been observed.

3.1 Choice of Pairs R_i, S_j

Clearly it is not useful to update with a pair of relations (R_i, S_j) whose codomains hardly match each other. The possible pairs of relations are ordered according to the maximum probability of matching (from the least prejudiced distribution), averaged over all elements of the codomain C_i :

$$AvMatch(h_{ij}) = \frac{\sum_{x \in C_i} \max_{y \in D_j} (Pr(y \in LPD(h_{ij}(x))))}{|C_i|} \tag{8}$$

We use this in preference to the average maximum probability of matching, defined as

$$AvMaxMatch(h_{ij}) = \frac{\sum_{x \in C_i} \max_{y \in D_j} (Pr(y \in h_{ij}(x)))}{|C_i|} \tag{9}$$

because the latter measure is not necessarily helpful if there is a large amount of uncertainty in the approximate mapping. For example, if

$$C_i = \{c_1, c_2\}$$

$$D_j = \{d_1, d_2, d_3\}$$

then the universal matching function

$$h1_{ij}(c_1) = \{d_1/1, d_2/1, d_3/1\}$$

$$h1_{ij}(c_2) = \{d_1/1, d_2/1, d_3/1\}$$

has an $AvMaxMatch$ of 1 (since it definitely links every element of C_i to *something* in D_j), whereas

² In common with many uses of naive Bayes in machine learning and information retrieval, the independence assumption is hard to justify theoretically but appears to be valid in practice.

$$h2_{ij}(c_1) = \{d_1/0.9\}$$

$$h2_{ij}(c_2) = \{d_2/1, d_3/0.4\}$$

would only have an *AvMaxMatch* of 0.95 although it is much more specific than *h1_{ij}*.
 For the cases above

$$AvMatch(h1_{ij}) = 1/3$$

$$AvMatch(h2_{ij}) = 0.85$$

This discriminates against matching functions *h_{ij}* that are very unspecific.

It also makes little sense to choose *A* and *B* unless they are close to being key domains i.e. uniquely identifying an object. This can be estimated by looking at the cardinality of the domain.

3.2 Class Matching

Having determined equivalent instances from the two sources, we can look for correspondences between the different classification structures. For example, online music sources are typically organised hierarchically, but one site’s

music > rock > classic 1970’s

section may correspond (or correspond mostly) to another’s

music > rock&pop oldies

Ideally, it should be possible to map such categories into a user’s personal hierarchy – here, we concentrate on extracting rules from the overlap between categories in different classification structures based on a sample; we then use the derived rules to predict likely categorisations of new examples.

4 Application to Test Databases

Two test cases have been investigated; the first uses the instance-matching only, and the second uses instance matching and genre mapping to improve performance.

4.1 Results on News Data

The SOFT algorithm was applied to 1,701 stories from the BBC news archive (www.solutionseven.co.uk/bbc/ between May 19 and 31, 2005 inclusive) and 552

BBC	
Headline	Oldest FA Cup sells for £420,000
Original Publication-Date	2005/05/19 19:51:30
Description	The oldest existing version of the FA Cup becomes the world’s most expensive piece of football memorabilia.
Content	“The oldest existing version of the FA...”

SKY	
Title	FA CUP FETCHES £478,000
Last Updated	12:17 UK, Thursday May 19, 2005
Description	The oldest surviving FA Cup has been sold at auction for £478,400 - a world record for an item of football memorabilia.
Content	“Made in 1896 as a replacement for the stolen original, it...”

stories from the Sky news archive (www.sky.com/skynews/archive, same period). For comparison, a manually produced ground truth established 353 true matches - this figure is low due to different editorial styles and content in these two companies. An example is shown below. Using the SOFT algorithm, we find 380 pairs of matching news stories when the similarity threshold between two stories is set to 0.15. Figs 1 and 2 illustrate the SOFT result, a recall of 100% and an average precision of 92%.

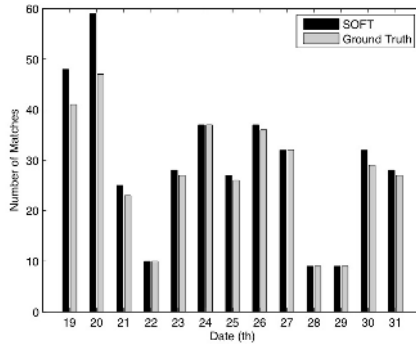


Fig. 1. The number of identified matches in the two data sources and the ground truth

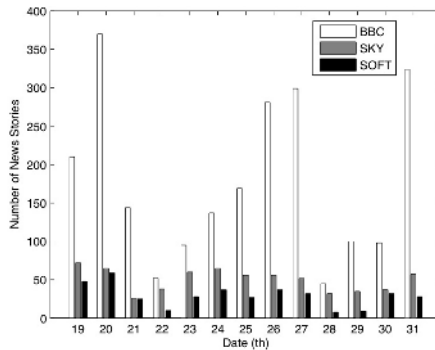


Fig. 2. The number of identified matches in the two news data sources compared to the total numbers of stories from the two sources

4.2 Instance and Genre Matching

The two film websites “rotten tomatoes”³ and the Internet Movie Database⁴ are “user-maintained” datasets which aim to catalog movie information. The databases denoted *dbT* and *dbI* below are derived from these sites, respectively containing 94,500 and 94,176 film records, were used in SOFT fusion experiments. Since *dbT* and *dbI* are produced by two different movie web sites, there is inevitable “noise” existing in the film data; i.e. different tag sets, different genre names and missing elements.

³ www.rottentomatoes.com

⁴ Information courtesy of Internet Movie Database (www.imdb.com). Used with permission.

dbI	
Title	Gentleman B.
Year	2000
Directed_by	Jordan Alan
Genre	Thriller
Aka	Gentleman Bandit, The (2000) (USA: MIFED title)
Country	USA

dbT	
Title	Gentleman Bandit
Year	2000
Director	Director: Jordan Alan
Genre	Genre: Dramas, Film Noir, Blackmail
MPAA_rating	NOT Rated
Cast	Cast: Ed Lauter, Peter Greene, Justine Miceli, Ryan O'Neal,

In order to match attributes, some very simple string matching functions were used as follows:

- (i) String S_1 is an approximate substring of S_2 if S_1 is shorter than S_2 and most words in S_1 are also in S_2 .
- (ii) String S_1 is an approximate permutation of S_2 if they have a high proportion of common words, i.e. degree of match = proportion of common words, which must be at least two (typical strings are only a few words long).

Both ignore “stop” words such as the, and, etc. We note also that it is possible to obtain better results for people’s names (attributes such as *cast*, *director*, etc) using a more structured approach which extracts first name and surname and then matches on that basis.

The average matches between domains are given in Table 1.

Table 1. Average degree of match between attributes in *dbI* and *dbT*

dbI attributes	dbT attributes	average matching using most probable element in least prejudiced distribution
Year	Year	100%
Title	Title	41%
Directed_by	Director	27%
Aka	Title	21%

On the basis of the three attributes, the system identified movies from *dbI* dated 1976-1990 which were also in *dbT*, and compared the genre classification. The similarity threshold between two film records was set to 0.5 giving a total of 14,124 movies which are found to be identical.

The similarity between two genres is relatively hard to decide from text string matching. For example, “animation” is not similar to “children” from the point of view of text matching, but the extension of the sets of films in these two categories shows considerable overlap. Some examples of interesting genre mappings are listed in Table 2.

Table 2. Examples of matching genre pairs determined by the system

dbT genre	dbI genre
Animation	Children
Comedy	Drama
Horror	Suspense
Sci-fi	Fantasy

4.3 Results on Unseen Data

The attribute and genre mappings were applied to a new set of 24,839 entries from *dbI* (calendar years 2000-05), trying to find matches in *dbT*. For comparison, a manually produced ground truth established 1274 true matches – this figure is low due to the relatively large number of entries for TV series, “foreign” movies etc in *dbI* which are not included in *dbT*. Using the SOFT algorithm without genre mapping, we find

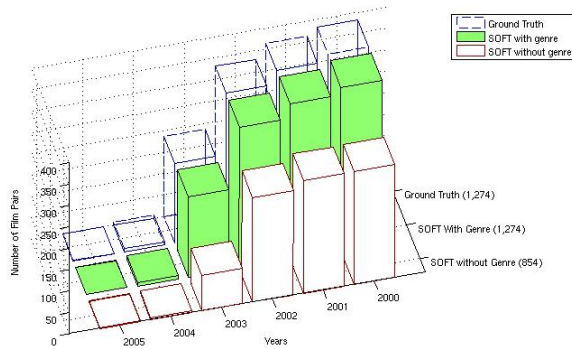


Fig. 3. The number of correctly identified matches in the two data sources, without and with genre mapping

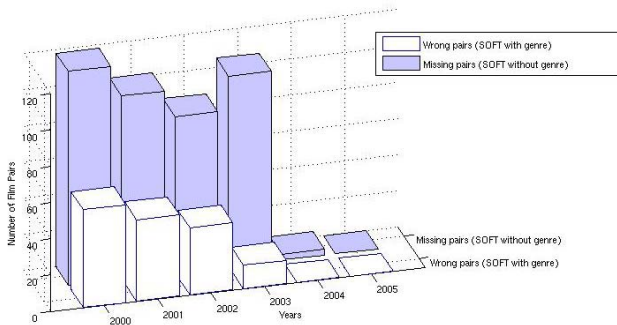


Fig. 4. The number of incorrectly identified matches when using genre information (white blocks) and the number of pairs not identified when not using genre information (shaded blocks)

861 pairs of matching film entries when the similarity threshold between two films is set to 0.44. With the presence of the ground truth, 261 film matching pairs out of 382 film pairs in 2000 are missing, 102 out of 364 in 2001 are missing, 87 out of 330 in 2002 are missing, 60 out of 142 in 2003 are missing, and 3 out of 8 in 2004 are missing (see figure 1). This represents a recall of 67 % and a precision of 100%. Incorporating the genre mapping as well produces a much better (100%) recall, with a slight loss in precision – see figures 3 and 4.

5 Summary

We see multimedia retrieval as highly dependent on the presence of good meta-data, and have identified a problem when different sources use different meta-data schemata. We have presented a method for finding similar attributes within such schema, and identifying similar categories.

Initial results are promising, although further testing is necessary to fully establish the validity of this method, involving better attribute-matching functions and a wider variety of datasets. In future we also intend to investigate the use of this method with a personal hierarchy, so that the user can adapt source meta-data to his/her own preferences and hence improve the retrieval process.

Appendix - Mass Assignment Theory

Mass assignment theory is covered in [1, 2]. We give a brief outline for completeness. Given a finite, discrete universe U , a mass assignment on U is a probability distribution over the power set of U (i.e. a random set [19]). A mass assignment is similar to the basic probability assignment of Shafer Dempster theory [20], but allows mass on the empty set. A mass assignment defines a family of probability distributions, and can be related to a fuzzy set as shown by the following example.

We ask a set of people to define the dice values they will accept as *small*, and find that the probability distribution on the set of values is:

$$\{1\} : 0.1, \{1, 2\} : 0.6, \{1, 2, 3\} : 0.3$$

(i.e. 10% will only accept 1 as *small*, 60% accept 1 or 2 etc).

The membership function of the fuzzy set *small* for any element is defined as the proportion of voters who accept that value as satisfying their definition of *small* i.e. the fuzzy set is $\{1 / 1, 2 / 0.9, 3 / 0.3\}$ where notation $a / \mu(a)$ indicates that element a has membership $\mu(a)$ in the set.

The *least prejudiced distribution* is a probability distribution on U obtained by sharing mass equally among the elements of each set. This corresponds to the probability of a dice score being chosen if we select a person from the voters at random and then ask that person to select one value from their possible set of accepted values when told the dice value is *small*.

In the case above, assuming a uniform prior we obtain the least prejudiced distribution:

$$1 : 0.1 + 1/2(0.6) + 1/3(0.3) = 0.5$$

$$2 : 1/2(0.6) + 1/3(0.3) = 0.4$$

$$3 : 1/3(0.3) = 0.1$$

By reversing this process we can work backwards from an observed distribution (assumed to be the least prejudiced distribution) to a mass assignment and fuzzy set.

Acknowledgements

This work was partially supported by BT Intelligent Systems Lab (www.btexact.com) and by the FP6ePerSpace project IST 506775 (www.ist-eperspace.org).

References

1. Baldwin, J.F., *The Management of Fuzzy and Probabilistic Uncertainties for Knowledge Based Systems*, in *Encyclopedia of AI*, S.A. Shapiro, Editor. 1992, John Wiley. p. 528-537.
2. Baldwin, J.F., T.P. Martin, and B.W. Pilsworth, *FRIL - Fuzzy and Evidential Reasoning in AI*. 1995, U.K.: Research Studies Press (John Wiley). 391.
3. Berlin, J. and A. Motro, *Autoplex: Automated Discovery of Content for Virtual Databases*. Springer LNCS 2172, 2001: p. 108-122.
4. Bilenko, M., R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, *Adaptive Name Matching in Information Integration*. Ieee Intelligent Systems, 2003. **18**: p. 16-23.
5. Chang, K.C.C. and H. Garcia-Molina, *Approximate Query Mapping: Accounting for Translation Closeness*. VLDB Journal, 2001. **10**(2-3): p. 155-181.
6. Chang, K.C.C., H. Garcia-Molina, and A. Paepcke, *Boolean Query Mapping Across Heterogeneous Information Sources*. Ieee Transactions on Knowledge and Data Engineering, 1996. **8**(4): p. 515-521.
7. Dey, D., S. Sarkar, and P. De, *A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases*. Ieee Transactions on Knowledge and Data Engineering, 2002. **14**(3): p. 567-582.
8. Ding, Y. and S. Foo, *Ontology research and development. Part 2 - a review of ontology mapping and evolving*. Journal of Information Science, 2002. **28**(5): p. 375-388.
9. Doan, A., P. Domingos, and A. Halevy, *Learning to Match the Schemas of Data Sources A Multistrategy Approach*. Machine Learning, 2003. **50**(3): p. 279-301.
10. Elfeky, M.G., V.S. Verykios, and A.K. Elmagarmid. *TAILOR: A Record Linkage Tool Box*. in *Proc International conference on data engineering*. p. 17-28. 2002. San Jose, CA: IEEE Computer Society.
11. Fellegi, I.P. and A.B. Sunter, *A Theory for Record Linkage*. J. American Statistical Assoc, 1969. **64**: p. 1183-1210.
12. Gal, A., A. Trombetta, A. Anaby-Tavor, and D. Montesi. *A Model for Schema Integration in Heterogeneous Databases*. in *Proc Seventh International Database Engineering and Applications Symposium (IDEAS'03)*. p. 2-11. 2003. Hong Kong: IEEE Press.
13. Madhavan, J., P.A. Bernstein, P. Domingos, and A.Y. Halevy, *Representing and Reasoning about Mappings between Domain Models*. Proceedings of the National Conference on Artificial Intelligence, 2002: p. 80-86.
14. Madhavan, J., P.A. Bernstein, and E. Rahm, *Generic Schema Matching with Cupid*. Proceedings of the International Conference on Very Large Data Bases, 2001: p. 49-58.

15. Martin, T.P., *Searching and Smushing on the Semantic Web - Challenges for Soft Computing*. Studies in Fuzziness and Soft Computing, 2004. **139**: p. 167-186.
16. Martin, T.P., *Soft Integration of Information with Semantic Gaps*, in *Fuzzy Logic and the Semantic Web*, E. Sanchez, Editor. 2005, Elsevier.
17. Newcombe, H.B., J.M. Kennedy, S.J. Axford, and A.P. James, *Automatic Linkage of Vital Records*. Science, 1959. **130**: p. 954-959.
18. Rahm, E. and P.A. Bernstein, *A Survey of Approaches to Automatic Schema Matching*. The VLDB Journal, 2001. **10**: p. 334-350.
19. Nguyen, H.T. *On Random Sets and Belief Functions*, J. Math. Anal. & Appl., 1978, **65** pp.531-542
20. Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press 1976.

Author Index

- Albers, R. 112
- Bade, Korinna 91
- Bartolini, Ilaria 222
- Bergler, Sabine 65
- Bouamrane, Matt-Mouley 79
- Boughanem, Mohand 44
- Bruno, Eric 128, 168
- Desurmont, X. 112
- Detyniecki, Marcin 55
- De Luca, Ernesto W. 91
- de Vries, Arjen P. 180
- Djordjevic, Divna 19
- El Demerdash, Osama 65
- Ferrane, Isabelle 141
- Fontijn, Willem 102
- French, James 191
- Ibrahim, Zein Al Abidin 141
- Izquierdo, Ebroul 19
- Jaspers, E. 112
- Jin, Xiangyu 191
- Joly, Philippe 141
- Jose, Joemon M. 207
- Kosseim, Leila 65
- Langshaw, P. Karen 65
- Lee, Hyowon 155
- Loiseau, Yannick 44
- Lukkien, J. 112
- Luz, Saturnino 79
- Marchand-Maillet, Stéphane 128, 168
- Marchionini, Gary 35
- Marteau, Pierre-François 236
- Martin, Trevor P. 266
- Ménier, Gildas 236
- Michel, Jonathan 191
- Moenne-Loccoz, Nicolas 128, 168
- Nesvadba, Jan 102, 112
- Nürnbergger, Andreas 91
- O'Connor, Noel E. 155
- Omhover, Jean-Francois 55
- Palo, J. 112
- Pietarila, P. 112
- Popovici, Eugen 236
- Prade, Henri 44
- Ramírez, Georgina 180
- Rifqi, Maria 55
- Rüger, Stefan 1
- Sav, Sorin 155
- Shen, Yun 266
- Shevade, Bageshree 251
- Sinitsyn, Alexander 102, 112
- Smeaton, Alan F. 155
- Stober, Sebastian 91
- Sundaram, Hari 251
- Truyen, R. 112
- Urban, Jana 207
- Westerveld, Thijs 180
- Wijnhoven, R. 112