ERCIM-97-W004
CNR

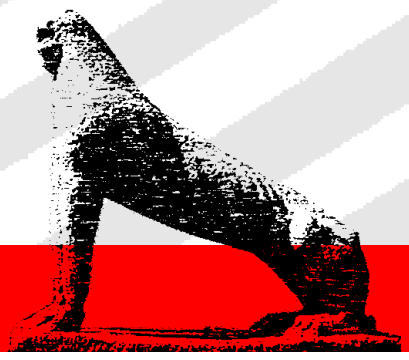
European Research Consortium
for Informatics and Mathematics

ERCIM

Fourth DELOS Workshop

Image Indexing and Retrieval

San Miniato, Italy, 28-30 August 1997



INTRODUCTION

Images have always been a fundamental communication medium. The impressive achievements in information technology of the last decade have brought images to play a central role in computerized information systems, thus opening an enormous application area. The amount of images stored and accessed in electronic form is already very large, and is rapidly increasing. It is expected that images will be a major content resource of digital libraries.

In this context, the ability to select from a large collection those images that are relevant to a user request, specified either in visual or in textual terms, is at the heart of the image retrieval problem, and, at the same time, a fundamental functionality of a modern information retrieval system. Queries expressed in visual terms (e.g., images themselves or elements thereof) are handled on the basis of similarity criteria, and thus aim at providing the user with images that "look like" the query; this is "content-based image retrieval", a form of retrieval currently being studied by the research community. On the other hand, queries expressed in textual terms, such as natural language sentences or formal expressions using a controlled vocabulary with a possibly complex syntax, address the content of the images; this is known as "semantic content-based image retrieval".

The fourth workshop of the DELOS working group addressed the topic of "Image Retrieval" and was hosted by CNR in San Miniato, a small medieval town not far from Pisa, from 28-30 August, 1997. DELOS is a Working Group funded by the IT Long Term Research programme of the European Commission to study and investigate existing and emerging technologies and issues relevant to digital libraries.

The Workshop brought together 20 participants from eight different European countries, as well as 6 invited speakers, one from Europe and 5 from the United States. The workshop thus provided a significantly broad picture of the research on image retrieval under way both in Europe and in the US.

The invited speakers' presentations addressed different aspects of the image retrieval problem. In particular, Alberto Del Bimbo, from the University of Florence, gave a comprehensive overview of content-based image retrieval. Fang Liu from the MIT Media Lab illustrated a number of digital library projects completed or in progress in the Vision and Modeling Group at the MIT Media Laboratory. David Forsyth from the University of California at Berkeley presented the results of a research project aiming at the retrieval of pictures on the basis of the objects depicted. Simone Santini from the University of California at San Diego investigated the problem of meaning in image databases, arguing that the meaning of an image should be considered as a result of the interaction between the user and the database, rather than as an intrinsic property of the images. John Smith from the Image and Advanced Television Laboratory of Columbia University presented a new approach for automatically classifying images using both image features and related text. He argued that the synergy of textual and visual information in Web documents provides a great opportunity for improving the image indexing and searching capabilities of Web image search engines. Olivia Frost from the University of Michigan presented a project under way at her university aimed at exploring the development and evaluation of multimode retrieval schemes which employ both image and text and which are multiple path, iterative and user-directed.

Other presentations from the European researchers ranged from modeling, to the study of efficient access methods supporting similarity retrieval, feature extraction methods and applications. In particular, Carlo Meghini (IEI-CNR Pisa) presented an approach for combining syntax and semantics in image retrieval, while Youssef Lahlou (GMD) described an object-oriented model for semantic image indexing. The presentation of Roger Weber (ETHZ Zurich) and Pavel Zezula (IEI-CNR Pisa) illustrated the problems related to the definition of access structures for similarity retrieval in high dimensional data spaces. Specific solutions were proposed. Roger Mohr (INRIA) presented an approach to image retrieval that uses local characteristics; Ines Oliveira (INESC Lisbon) illustrated a comprehensive set of image processing techniques for video content extraction. Finally, Andre Csillaghy (ETHZ Zurich) presented an application of image browsing and retrieval, applied to astronomical archives.

These proceedings include extended abstracts of the talks given at the Workshop.

We should like to that ESPRIT and ERCIM for supporting the Workshop, and our collegue Tarina Ayazi, as well as the local people at San Miniato, to make it happen.

Carlo Meghini
Pasquale Savino

Visual Querying by Color Perceptive Regions

A. Del Bimbo, P. Pala
Dipartimento di Sistemi e Informatica
Università di Firenze
50139 Firenze, Italy

Extended Abstract

Introduction

Image databases are now currently employed in an eclectic range of different areas such as entertainment, art history, advertising, medicine and industry among others. In all these contexts, a key problem regards the modality of accessing visual image content. The visual content of an image should be intended as what humans remember after looking at it for a couple of minutes. It may include shapes of relevant objects, color distribution or color patches, textured surfaces, or the arrangement of visual elements on the image plane. In image databases, unlike textual and numerical databases, the visual content by itself has no ordering rules, unless textual labels are associated with images. Conventional attempts to cast visual features into textual keywords [Srihari] has been far recognized to be inadequate in indexing pictures. The minor expressiveness of text with respect to visual features doesn't allow to fully exploit capabilities of human memory. Items retrieved through a textual query could not be relevant at all for user's expectation.
This is the reason why retrieval based on visual content has been identified as the means to overcome this modal clash. Pattern recognition and image analysis algorithms are used to extract parameters of the visual content and filter out unrelevant images. The relevance of visual elements for the final user depends on his subjectivity (which is not known in advance, when the database is created) and on the context of application (which is known, instead). On the one hand, user's subjectivity implies that modules are employed which are able to cope with imprecision and lack of knowledge about image content. On the other hand, the fact that the context of application is known in advance, can drive the choice of pattern recognition algorithms and functions that must be included in the system. For example, spatial arrangement is relevant to a doctor examining a heart section; color arrangement is important when looking at paintings, but the presence of particular color tones or the disposition of color patches on the canvas will be better remembered by an expert than by an occasional user.

Generally speaking, an efficient system for retrieval by visual content should be able to:
- provide a query paradigm that allows users to naturally specify both selective and imprecise queries;
- define retrieval facilities that are meaningful for the context of application;
- define similarity metrics which are satisfactory for user perception.

Querying by visual example is the interaction paradigm that exploits human natural capabilities in picture analysis and interpretation. It requires that visual features are extracted from images and used as indexes in the retrieval phase.
Querying by visual example allows imprecision and incompleteness of expression, by letting users draw a sketch of their memory of the image (for example colored shapes arranged in a specified pattern). It must not be expected that the first answer be already satisfactory. Rather, a continuous interaction must be foreseen, through which
the original query of the user can be refined or changed on the basis of the retrieved pictures.

In recent years, prototype systems have been proposed which implement retrieval by content from image databases by addressing different facets of the informative contents of pictorial data, such as object texture organization, [Picard] shape similarity with visual sketches [Kato] , [Mehrotra], [DelBimbo97], semantic relationships between imaged objects [Chang-84], [Chang-87], [DelBimbo-95].
With the increasing availability of devices supporting acquisition and visualization of color images, a growing attention is also being focused on chromatic features as a key facet in image content.

In this presentation we discuss the PICASSO system, developed at the Visual Information Processing Lab of the University of Florence, Italy. The system is a complete framework providing facilities for image indexing and retrieval based on shapes, colors, and spatial relationships. Only color retrieval facilities are discussed in this paper. For shape and spatial relationship based retrieval the reader can refer to [DelBimbo-97] , [DelBimbo-96].

Concerning colors, the system supports both retrieval by global color similarity and retrieval by similarity of color regions. Shape, size, and position of color regions are considered as optional features that the user can select in the query.

Color-based Image Retrieval Experiences

Previous experiences in color--based retrieval essentially address two kinds of problems. The first regards finding database images whose color distribution is globally similar to that in a query image. In this problem, the user's interest lies on the chromatic content of the whole image: what is represented in a picture has no particular relevance. In painting databases, for example, this kind of query could help in finding paintings of the same author, or of a certain author's period, or perceptually similar paintings.
The second problem is finding a certain object in a complex scene, using its chromatic properties. The rest of the scene is not relevant for the user's purposes. Only the presence and location of the object are interesting.

Image retrieval based on global color distribution has been formerly proposed by Swain and Ballard [Swain]. Image color distribution is represented through color histograms, which are obtained by discretizing the color space, and counting how many pixels fall in each discrete color. Color histograms have the property of being invariant to translation and rotation. They slowly change when objects are partially occluded, or when there are changes in scale and angle of view. Retrieval is performed by evaluating the intersection between the global color histogram of the user--provided example and stored images.

The QBIC database system [QBIC], [Cody] evaluates similarity in terms of global properties of color histograms. A weighted distance measure is used to evaluate the similarity of color histograms. The distance measure is a weighted cross correlation between histogram bins. Weights represent the extent to which two histogram bins and are perceptually similar to each other.
In [Jain] A.K. Jain and A. Vailay used both color and shape features, analyzed over the whole image. In their system, the query is formulated through an example image and retrieval is accomplished by a similarity measure computed on the basis of the global color histogram and image edges.

Few authors have allowed retrieval of images based on similarity of color patches or objects. This is due to the inherent major complexity of this kind of retrieval. Searching for localized color regions, eventually corresponding to relevant objects, requires to use effective algorithms to locate uniform color regions, more complex data structures to represent color image properties, and more complex algorithms to evaluate perceptual similarity.
In [Tanaka] images are partitioned into blocks of equal size, each associated with its own local histogram. Similarity matching considers adjacency conditions among blocks with similar histograms. However, blocks are created according to a static partitioning of the image, which is generally inadequate to reflect the original arrangement of colors in a complex image.
In [Vinod] an iterative technique is proposed to identify image regions which can potentially represent a given object, based on color features. Color Histogram Intersection is used to evaluate the match between image regions and query objects. Regions of potential interest are extracted considering a square window and shifting it on the image by a fixed number of pixels in one direction at a time.

A different and more reliable solution requires that the whole image is segmented into homogeneous color regions. Chromatic and geometric features of such regions are matched against corresponding regions in the query model. Segmentation is typically the hardest problem to be solved. Matching of segmented images with query color patches or segmented objects is also difficult. In a query by example, the color patches sketched by the user correspond to his approximated view of the image searched, rather than to the true image patches. Shapes of color regions of database images, as resulted from the segmentation, could not fit shapes of regions specified in the query.
The PICASSO system presented in this paper exploits a hierarchical multi-resolution segmentation and graph matching in order to support effective retrieval based on color regions. The system has been designed mainly for content based retrieval of paintings and art images, in which the assumption of smooth changes of colors within homogeneous regions does not hold.

Hierarchical Color Image Segmentation

The number r of regions which are produced in the segmentation process determines the level of precision of the image partitionment. The value of r can be determined adaptively on the basis of statistical analysis of color distributions in the sampled color space [Corridoni] However, in the framework of an image retrieval system and in the absence of specific assumptions about images being stored, one such approach may lead to segmentations

which do not meet user's expectations about perceptual groups. Moreover, at storage time it is impossible to forecast the level of precision which will be requested in the detection of image properties expressed by users in specific queries.
If the segmentation process fails to detect a region in an image or performs a segmentation into regions which is different from that expressed by the user in the query, that image will not be retrieved in the searching process. Both these considerations evidence that, at storage time, the optimal level of precision cannot be defined.

In the PICASSO system, this hurdle is circumvented by creating multiple descriptions of each data, each one covering a different level of precision. Images are analyzed at different levels of resolution in order to obtain a pyramidal segmentation of color patches. Each region at level n is obtained by clustering adjacent regions at level n-1. A region energy is associated to each region. This energy is obtained as a weighted sum of three entries:
- the area;
- the color uniformity;
- the color contrast.

The image energy is defined as the sum of all region energies. Image segmentation is performed by minimizing a function such that the area is to be maximized, the color uniformity to be maximized and color contrast to be minimized.
Image segmentation is performed by iteratively updating region clusters at each resolution level, separately. At the lowest level of the pyramid, each region corresponds to a pixel in the image. Starting from the lowest level of the pyramid, two adjacent regions are searched whose merging would decrease the image energy. This procedure is recursively applied until the coarsest resolution is reached, such that the entire image is represented by a single region.

Color Region Representation

In order to support effective retrieval by content of paintings, a color space has been chosen, such that close distances in the color space correspond to close distances for the user perception. This condition, which is not exhibited by the RGB space, has been accomplished with the adoption of the perceptually uniform CIE L*u*v* color space. Close colors in this space correspond to perceptually close colors.
In our approach, a uniform tessellation of the L*u*v* space has been performed and the number of colors has been reduced to 128.

Region description

Color regions are modeled through their spatial location, area, shape, average color, and a binary 128-dimensional color vector. Entries in the color vector correspond to reference colors. If a reference color is present in the region, its corresponding entry in the color vector is set to 1 (0 otherwise).

At the coarsest resolution of its pyramidal representation, the image is represented by a single region with a color vector retaining the global color characteristics for the whole image. As the resolution increases, regions correspond to smaller area in the image and are thus characterized by the presence of a smaller number of reference colors. This yields an increasing localization of chromatic properties into smaller regions.

Color Image Retrieval

The PICASSO system supports retrieval by visual example of images with one or several colored regions. Queries may include one or more sketched color regions of any shape and in any relative position. Position, area, elongation, and orientation attributes can be selected as relevant features of a sketched region. Color regions are sketched either by drawing a region contour and then filling it with a color selected from a color picker, or by contouring a region of an image that has been previously answered to a query, or that is selected from a set of samples. At database population time, images are automatically segmented and modeled through a pyramid structure as expounded previously. The highest node of each pyramid includes both the binary color vector associated with the whole image, and the image color histogram. Image color histograms represent images global color appearance. They are used to compute the similarity between two images in terms of their global chromatic contents.
A color index file is built by considering color vectors of the highest nodes associated with all the database images. The color index file has 128 entries, one for each reference color. Each entry storing a list of database images where that reference color is present.

Given a query, the color index file is used, to select a set of candidate images that contain regions with the same colors as the query. Unrelevant images which do not contain some region with the same color as the

query are quickly filtered out. The pyramid structure of each candidate image is analyzed in order to find the best matching region R_I for each query region R_Q. Given a query region Q_R, the method used to find the best matching region of an image pyramid G can be summarized as follows:

procedure match (R_Q, G)
> Let V be the vertex of the pyramid G$
> Initialize the match M=0
> analyze (V, R_Q)
> return M
end
procedure} analyze(V, R_Q)
> Let C be the color of R_Q
> {\bf if} C is contained in the color vector of V
> Compute the match T between R_Q and V, according to MATCH)
> if T>M
> update M with T
> if V is a leaf
> stop
> else
> let V_1...V_K be the children of the current node
> for k=1 to K
> analyze(V_k,R_Q)
end

A similarity coefficient for the whole image is evaluated as the sum of scores of the best matching image regions for each query region, PICASSO system also supports retrieval by global color similarity. Querying by global similarity supports the user when he is interested in finding images with similar global chromatic contents, without having to localize them in a specific region in the image. Retrieval by global color similarity is carried out by evaluating the correlation between color histogram of the query image and that of database images. Experimental results are presented for a database of paintings.

Bibliography

[Chang-84] S.K. Chang and S.H. Liu.
Picture indexing and abstraction techniques for pictorial databases.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 6(4), July 1984.

[Chang-87]
S.K. Chang and C.W. Shi, Q. Y.Yan.
Iconic indexing by 2-{D} strings.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 9(3) pp. 413-427, July 1987.

[Cody] W.Cody.
Querying multimedia data for multiple repositories by content: The GARLIC project.
In Proc. on Visual Data Base Systems III, Lausanne, 1995.

[Corridoni] J.M. Corridoni and A. Del~Bimbo.
Multi-resolution color image segmentation.
In Dip. Sistemi e Informatica, Universita di Firenze Tech. Rep.,
no. 7-96, Jan. 1996.

[DelBimbo-95] A. Del Bimbo, E. Vicario and D. Zingoni.
Symbolic description of image sequences with spatio-temporal logic.
IEEE Trans. on Knowledge and Data Engineering,
(7)4 pp. 609--621, Aug. 1995.

[DelBimbo96] A. Del Bimbo, P. Pala.
Image Indexing Using Shape Based Visual Features
International Conference on Pattern Recognition, Wien, August 1996.

[DelBimbo97] A. Del Bimbo and P. Pala.
Visual image retrieval by elastic matching of user sketches.
IEEE Trans. on Pattern Analysis and Machine Intelligence.
(19)2, pp. 121-133, Feb. 1997

[Huang] C.L. Huang, T.Y. Cheng and C.C. Chen.
Color images' segmentation using scale space filter and Markov random field.
Pattern Recognition, (25)10 pp. 1217--1229, 1992.

[Jain] A.K. Jain and A.Vailaya.
Image retrieval using color and shape.
Pattern Recognition, 29(8) pp. 1233-1244, Aug. 1996.

[Kato] K.Hirata, T.Kato,
Query by Visual Example: Content-Based Image Retrieval.
In Advances in Database Technology - EDBT'92, A.Pirotte, C.Delobel, G.Gottlob (Eds.), Lecture Notes on Computer Science, Vol.580,

[Mehrotra] R.Mehrotra and J.E. Gary.
Similar-shape retrieval in shape data management.
IEEE Computer}, 28(9) pp. 57-62, Sept. 1995.

[Picard] F. Liu and R.W. Picard
Periodicity, directionality, and randomness: Wold features for image modeling and retrieval.
M.I.T. Tech. Rep., no. 320, 1995.

[QBIC] W.Niblack et alii,
The QBIC Project: Querying Images by Content Using Color, Texture and Shape
Res.Report 9203, IBM Res.Div. Almaden Res.Center, Feb.1993.

[Srihari] R.K. Srihari.
Automatic indexing and content-based retrieval of captioned images.
IEEE Computer, 28(9) pp. 49-56, Sept. 1995.

[Swain] M.J.Swain, D.H.Ballard,
Color Indexing
Int.Journal of Computer Vision, Vol.7, No.1, 1991.

[Tanaka] A. Nagasaka and Y. Tanaka.
Automatic video indexing and full video search for object appearances.
IFIP Trans., Visual Database Systems II, pp 113-127, Knuth, Wegner (Eds.), 1992. Elsevier.

[Vinod] V. Vinod, H. Murase,
Focussed Retrieval of Color Images
To appear on Pattern Recognition,

Image Retrieval Using Local Characterization

Cordelia SCHMID and Roger MOHR

INRIA
655 avenue de l'Europe
38330 Montbonnot
FRANCE
first.last-name@imag.fr

Abstract

This paper presents a general method to retrieve images from large databases using images as queries. The method is based on local characteristics which are robust to the group of similarity transformations in the image. Images can be retrieved even if they are translated, rotated or scaled. Due to the locality of the characterization, images can be retrieved even if only a small part of the image is given as well as in the presence of occlusions. A voting algorithm, following the idea of a Hough transform, and semi-local constraints allow us to develop a new method which is robust to noise, to scene clutter and small perspective deformations. Experiments show an efficient recognition for different types of images. The approach has been validated on an image database containing 1020 images, some of them being very similar by structure, texture or shape.

1 Introduction

Image retrieval is an important problem for accessing large image databases. We address the problem of retrieving any kind of images under the following conditions:

- partial visibility and complex background or clutter;
- different viewing angles,
- moderate changes in illumination;
- thousands of potential reference shapes.

Existing approaches use either geometric features of an object or rely on its luminance signature. Geometric approaches are robust to transformations and occlusions, but they only allow to deal with certain classes of objects. On the other hand, photometric approaches allow to deal with any kind of objects, but they do not work if the object is only partially visible. Furthermore, these methods are not invariant to any kind of image transformation. Recently [5] developped a method invariant to rotation using steerable filters. When considering colour, Slater and Healey [7] developped illumination invariant descriptors used for image recovery [2].

This paper presents a approach which allows to derive a set of greylevel or colour image invariant together with an indexation method allowing fast retrieval of parts of image already seen in similar condition (view point, illumination). The method uses local characteristics of the greyvalue signal which are invariant to similarity transformations. These characteristics are calculated at automatically detected keypoints, as shown in figure 1; for illustration purpose only some of the keypoints are displayed.

The originality of this work consists of several points. The use of local greyvalue differential invariants for indexing into a database presents the most important novelty. These invariants are continuous and

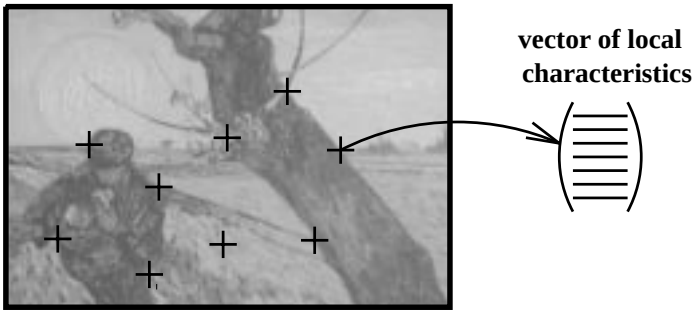


Figure 1: Representation of an image

independent of any image displacement. Another important point is the use of automatically detected keypoints which are representative of the object. Other authors [5, 9] use points fixed on a grid. As these grid points might not be significant, the vectors they use have to be much longer than ours. In case of occlusions, grid placement gets difficult and recognizing parts of images is impossible, as the grid can not be centered any longer. Our method avoids these drawbacks.

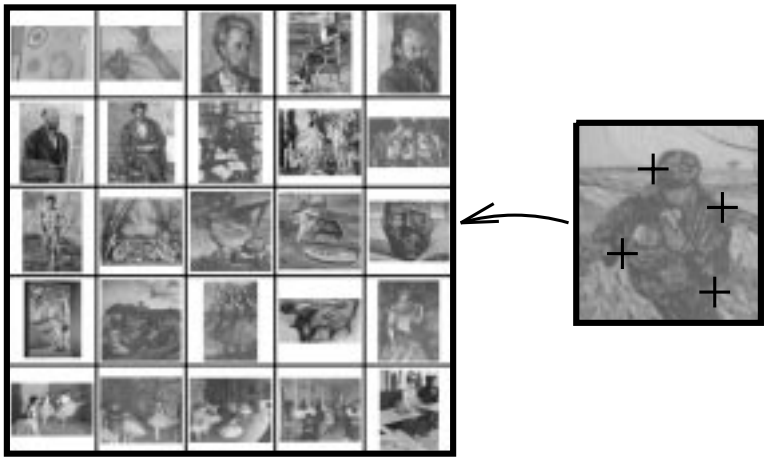


Figure 2: Research in the base, for illustration purpose only some of the keypoints are displayed

2 Method

The first step of our algorithm is the extraction of keypoints. The advantage of keypoints is that the informational content of the signal is high at their location. Furthermore, keypoints are local primitives. Standard vision algorithms exist for automatic extraction of keypoints. It is important that the detector is repeatable, that is results have to be invariant to image transformations. A comparison of different detectors in the presence of image rotation, scale change, light changes and image noise has shown a poor stability of existing methods and best results for the Harris detector. A stabilized implementation of this detector has been used in the present work.

The second step of our algorithm is to compute the local characterization. It is based on differential greyvalue invariants [3, 6] under the group of rigid motion. Due to a stable implementation of these invariants, a reliable characterization of the signal is obtained. A multi-scale approach [8, 4] makes this characterization robust to scale changes up to a factor 2. This has never been reported in the literature.

The third and final step of our algorithm is the retrieval or matching algorithm as schematized in figure 2. The image to be retrieved is compared to the images stored in the database. We therefore compared the vectors calculated for the image to be retrieved and the vectors calculated for the images in the database. The Mahalanobis distance is used to take into account uncertainties. A voting algorithm

determines the most likely image. To allow fast retrieval of the image, the vectors of the database are organized in an index table : the vectors are ordered in a multi-dimensional hash table. Each level of this multi-dimensional hash table indexes one component of a characterization vector.

If we are dealing with complex scenes the voting algorithm may result in several hypotheses. We therefore add constraints of local coherence. For a given match, at least half of the neighbor keypoints have to be compatible and angular spacements have to correspond. Robust recognition is then possible even in case of important geometric transformations and with only an image fragment.

3 Experiments

The database used for our experiments contains more than 1000 images. These images are of different types : painting images, aerial images and images of 3D objects. Some images of the database are shown in figure 3. Experiments conducted for this database have shown the robustness of the method to image rotation, scale change, partial visibility and scene clutter.

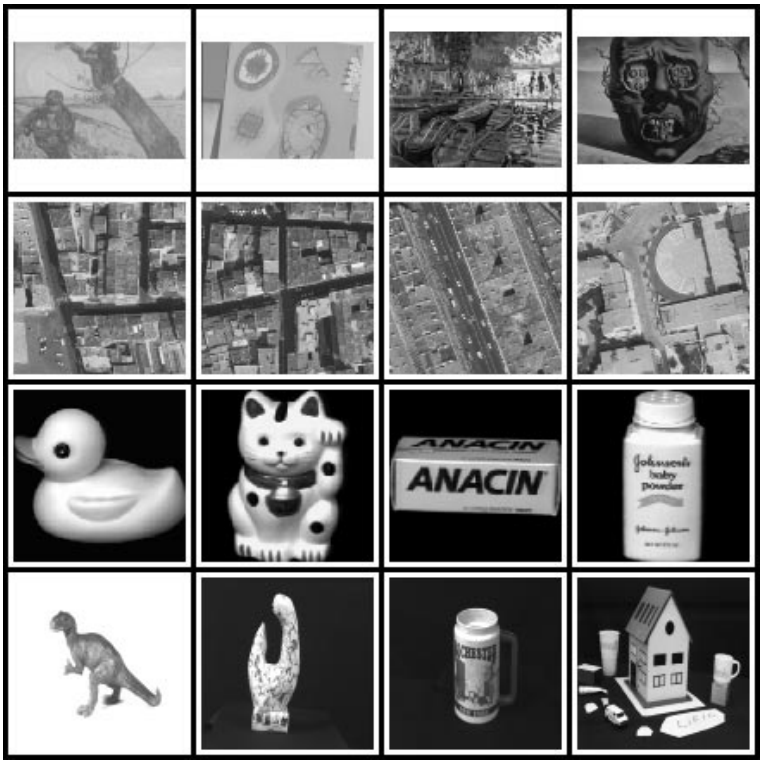


Figure 3: Some images of the database. The database contains more than 1000 images.

A set of test images to be retrieved contains 1000 images, either taken from a different point of view, under image rotation or scale change. The recognition rate obtained is 99%. Recognition for part of images has also been tested. The extracted parts cover 20 % or less of the entire image. The recognition rate is again near 100 % even if the image is rotated or scaled or if small viewpoint changes occur.

For more important viewpoints changes, for which we are able to recognize correctly being given the entire image, recognizing part of images works not as well (74%). This is due to the fact that small parts do not contain enough points, that is the number of votes is limited. In this case the robust algorithm can not overcome the uncertainty statistically.

Figure 4 shows parts of painting images which allow to correctly retrieve the entire image. Correct retrieval is also possible in case of image rotation and scale change for entire images as well as for parts of images (cf. 4).

Figure 4: Parts of paintings.

Another example is displayed in figure 6 for an aerial image. On the right of figure the recognized image is shown (a black frame indicates the corresponding part of the image). Recognition is possible for a part which has been rotated and scaled. Furthermore, there is a small perspective deformation between the two images, as the airplane has moved.

Typical retrieval time is about 20 second on a Spark10 for this kind of experiments: 15 seconds for image processing and 5 seconds for searching through the huge 20 Mb index table encoding the 1020 images. Faster implementations are easy to consider, either using devoted image processing systems, or by exploiting the large extrinsic parellism of the processes.

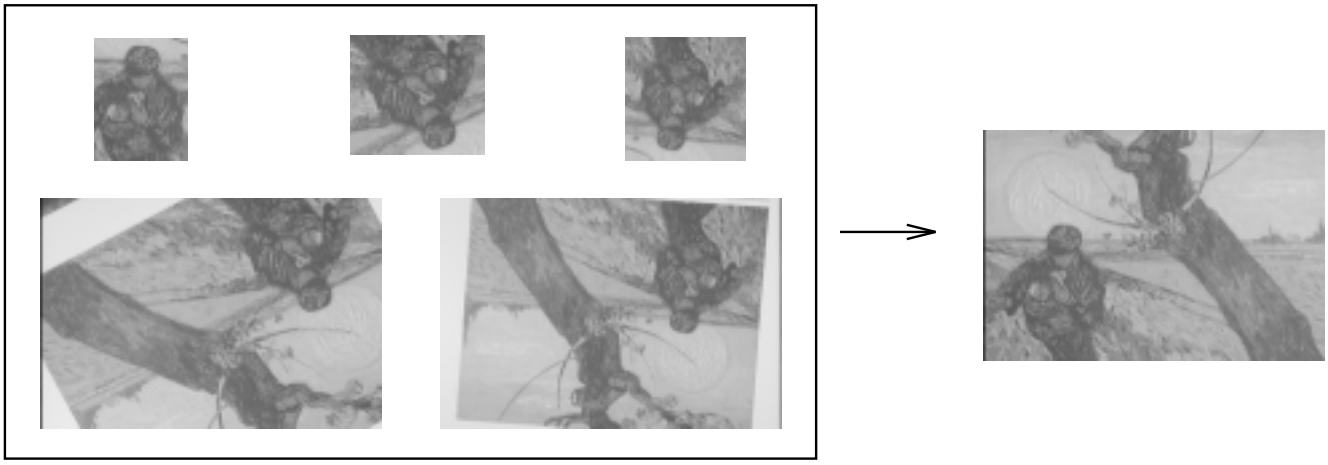


Figure 5: The image on the right is correctly retrieved using any of the images on the left.

4 Conclusion and future work

Our approach is an important contribution to image retrieval. It makes retrieval possible for images in situations which could not be dealt with before. We can identify images in case of partial visibility, image transformations and complex scenes. The success of our approach is based on the combination of differential invariants computed at keypoints with a robust voting algorithm and semi-local constraints. These invariants can be implemented with a sufficiently small filter size to capture local discriminant greylevel information. Moreover, the multi-scale approach makes our method robust to scale changes up to a factor 2.

We are presently extending these invariance to different directions:

- robust invariant in order to allow partial occlusion even in the local subwindow where the computation is done;
- colour invariance; instead following the path opened by [7], we experimented the colour invariance introduced by Funt and Finlayson [1]; in particular, this invariant seems very promising and it allows to some extend to take into account the shadows.

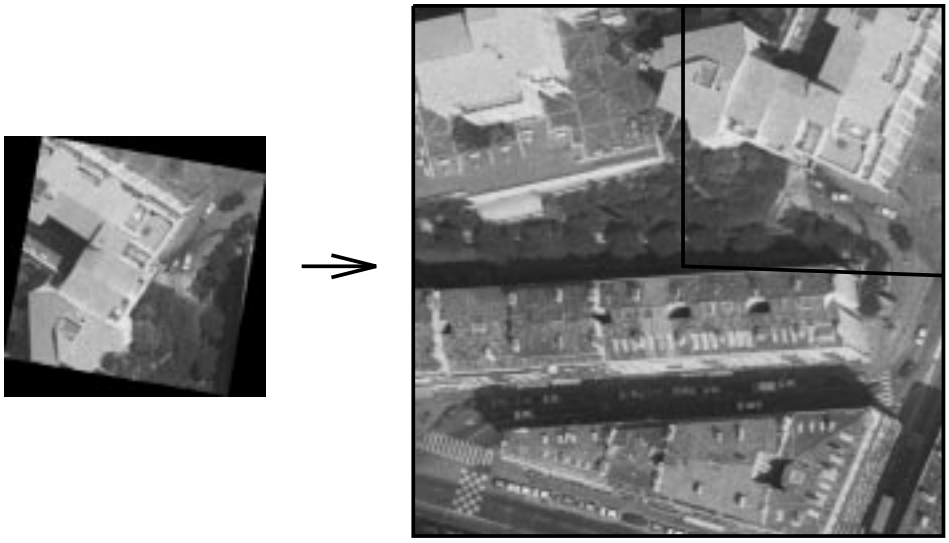


Figure 6: Recognizing part of an aerial image (courtesy of Istar).

5 Acknowledgements

Cordelia Schmid was supported by the European Community within the framework of the Human Capital and Mobility program.

References

[1] G. Finlayson, S. Chatterjee, and B. Funt. Color angular indexing. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, pages 16–27, 1996.

[2] G. Healey and D. Slater. Computing illumination-invariant descriptors of spatially filtered color image regions. IEEE *Transactions on Image Processing*, 6(7):1002–1013, 1997.

[3] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[4] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[5] R.P.N. Rao and D.H. Ballard. Object indexing using an iconic sparse distributed memory. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 24–31, 1995.

[6] B.M Romeny, L.M.J. Florack, A.H. Salden, and M.A. Viergever. Higher order differential structure of images. *Image and Vision Computing*, 12(6):317–325, 1994.

[7] D. Slater and G. Healey. Combining color and geometric information for the illumination invariant recognition of 3D objects. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 563–568, 1995.

[8] A.P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, Germany*, pages 1019–1023, 1983.

[9] X. Wu and B. Bhanu. Gabor wavelets for 3D object recognition. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 537–542, 1995.

Image and Video Modeling and Understanding

Fang Liu
The MIT Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
fliu@media.mit.edu

Abstract

Seven digital library related projects conducted in the Vision and Modeling Group of the MIT Media Laboratory are reviewed. These projects address a large variety of issues that are essential to building sophisticated, efficient, and user friendly tools for image and video library applications. The problems on which these projects focus include feature extraction, feature combination, similarity comparison, image and video understanding, and learning from man-machine interaction.

1 Introduction

This talk reviews seven digital library related projects completed or in progress in the Vision and Modeling Group of the MIT Media Laboratory. These projects address a large variety of issues that are essential to building sophisticated, efficient, and user friendly tools for image and video library applications.

In the existing image and video database applications, color and texture are the most commonly used features. The first work presented is a texture model that provides perceptually sensible image features. This model has been applied to both spatial and temporal texture modeling for image retrieval and video analysis.

Using low-level image features to achieve high-level image understanding is a challenging problem. The second project reviewed is a scene classification system, which successfully uses both color and texture information to classify indoor and outdoor consumer photographs.

Face recognition is a classic digital library application. The third project presented is a face detection and recognition system that uses visual learning. This system placed first in the 1996 FERET contest.

The ability to track and interpret human action is very important for video analysis and understanding. Three projects in this area are presented: Pfinder, American Sign Language recognition, and discourse video analysis. Pfinder provides a means to extract the gesture information from a video stream. The gesture information is then used to interact with artificial life or to interpret American Sign Language. The discourse video analysis algorithm uses both gesture and audio information to detect semantic patterns in monologue videos. This system can pick out stand-up comedians' punch lines!

Most of the existing retrieval systems require users to select similarity measures alone with a query. Relevance feedback is a more natural form of man-machine interaction. The last project presented is the FourEyes system. This image browser learns continuously from user feedback and incorporates a variety of models for representing the content of image and video.

Below is an extended abstract of the texture modeling work. Short descriptions of the other six projects are also provided. Related papers can be found at "http://www.media.mit.edu/vismod/", under "Publications".

2 Modeling Spatial and Temporal Textures (Liu and Picard)

2.1 Overview

Image texture features have been widely used in digital library applications. For image retrieval, a computer system is expected to return to its user database images that resemble the visual properties of the prototypes. To build such a system, it is important that the computational features used for pattern comparison are faithful to those used by humans in comparing patterns.

A random field decomposition theory named after statistician H. Wold allows the decomposition of a homogeneous texture pattern to be decomposed into three orthogonal components: harmonic, evanescent,

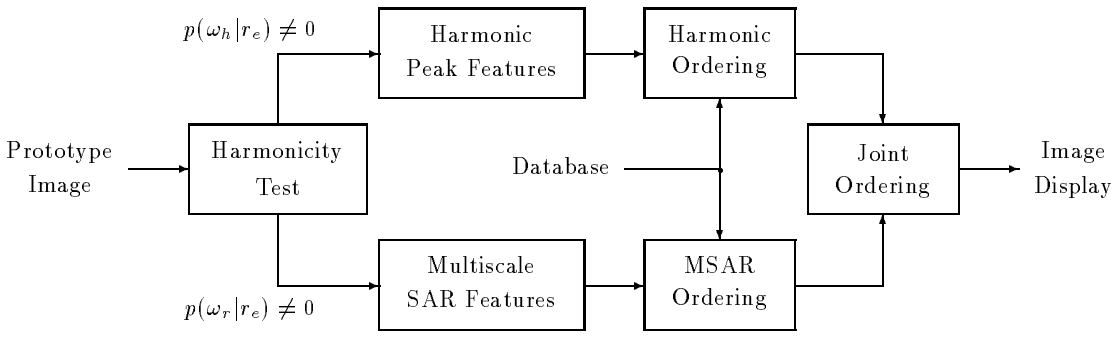


Figure 1: Flow-chart of the image retrieval system based on Wold texture modeling.

and indeterministic. The perceptual properties of the components can be described respectively as "periodicity", "directionality", and "randomness", agreeing closely with that of the top dimensions of human texture perception [13]. Hence, perceptually salient features can be constructed based on the Wold theory.

A textured image database retrieval system has been developed. The core of the system is a Wold-based shift, rotation, and scale invariant texture model. When compared to two other texture models, the Wold model appears to offer perceptually more satisfying retrieval results.

To investigate the perceptual properties of the Wold texture models, a psychophysical study was conducted. A highly significant correlation was found between the human and computer texture ranking data, suggesting that the component energy resulting from the 2-D Wold decomposition of an image is a good computational measure for the most salient dimension of human texture perception, the dimension of repetitiveness vs. randomness.

Applying the principle of Wold-based texture modeling to the temporal dimension, an algorithm is developed to simultaneously detect, segment, and characterize spatiotemporal periodicity. This technique is robust to noise and computationally efficient, providing a useful tool for video analysis and understanding.

In the following subsections, the Wold-based texture modeling work is briefly described. Details can be found in [5] and [6].

2.2 Textured Image Database Retrieval

The textured image database retrieval system is based on Wold texture modeling. Given a texture pattern, its repetitive structure is represented by its spectral harmonic peaks, and its randomness modeled by a multiresolution simultaneous autoregressive (MRSAR) fitting. Shown in Figure 1, the retrieval system consists of four stages. Given a prototype image, its level of repetitiveness is first examined by a harmonicity test. The Wold features are then extracted to characterize the periodic and the random components of the image separately. Based on each type of features, the entire database is ordered according to the image similarity to the prototype. (The Wold features of the database images are pre-computed.) Finally, the database orderings are combined for final query return.

The retrieval system was evaluated on the Brodatz Texture Database. This database contains 1008 eight-bit gray scale images cropped from the Brodatz album [3]. Each page of the album contributes nine images.

2.2.1 Harmonicity Test

The harmonicity of a textured image (*i.e.*, the amount of repetitive structure in the image) is determined by examining the energy distribution of the image autocovariance function. The autocovariance energy of a highly structured texture has periodic concentration throughout the 2-D displacement plane, while that of a random-looking texture concentrates in the small-displacement region. The ratio between the autocovariance energy in the small-displacement region and the total energy (total sum of the absolute value of the autocovariance function) can be used as a measure of image harmonicity.

The autocovariance energy ratio r_e was computed for each image in the Brodatz database. The histogram of the ratios has a bi-modal structure. Gaussian assumptions were made to model the energy ratio data using an expectation and maximization (EM) procedure [6]. Denote the resulting classes as ω_h (harmonic) and ω_r (random). The EM procedure gives the posterior probabilities $P(\omega_h|r_e)$ and $P(\omega_r|r_e)$, which can be used as the confidence measure of characterizing the image as highly structured and relatively unstructured, respectively.

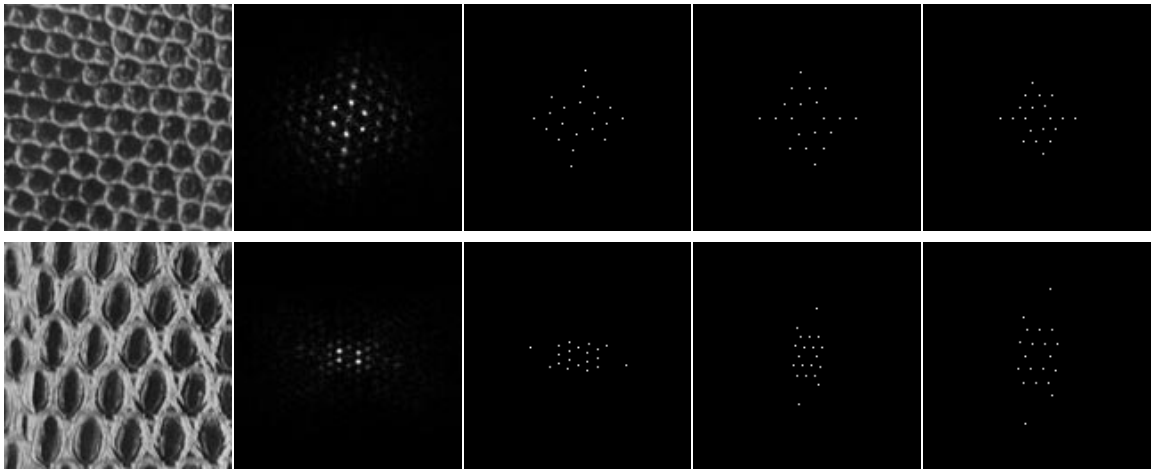


Figure 2: Harmonic peak feature rotation, and scale invariance. Top row: Reptile skin. Bottom row: Lizard Skin. Although the original patterns are in different scale and have relative rotation, their harmonic peak features allow rotation and scale invariant similarity comparison.

2.2.2 Feature extraction

In the second stage, the spectral harmonic peak features and the MRSAR features of the prototype image are estimated.

The harmonic peak feature set consists of the frequencies and the magnitudes of ten largest spectral harmonic peaks. When extracting the spectral peaks, the harmonic relationship among the peak frequencies is explicitly examined.

The harmonic peak features inherit from the Fourier spectral magnitude the property of spatial shift-invariance. To provide the ability of comparing images with respect to relative rotation and scaling, the peak feature set is rotated to align the lowest fundamental frequency to a chosen orientation (horizontal in this work) and scaled such that the distance between the lowest fundamental and the zero frequency is some chosen value (10 in this work). An example is shown in Figure 2.

The relatively unstructured texture components are characterized by using the MRSAR method introduced by Mao and Jain [8]. A second-order symmetric MRSAR model is fit to the image at three scales, resulting a 15-parameter feature vector and its covariance matrix.

2.2.3 Image Similarity Comparison

Two orderings of the entire database are generated in this stage. For each ordering, image similarities are measured by either the harmonic peak matching or the MRSAR feature Mahalanobis distances, and the database is sorted by the descending order of the image similarity to the prototype.

2.2.4 Joint Ordering

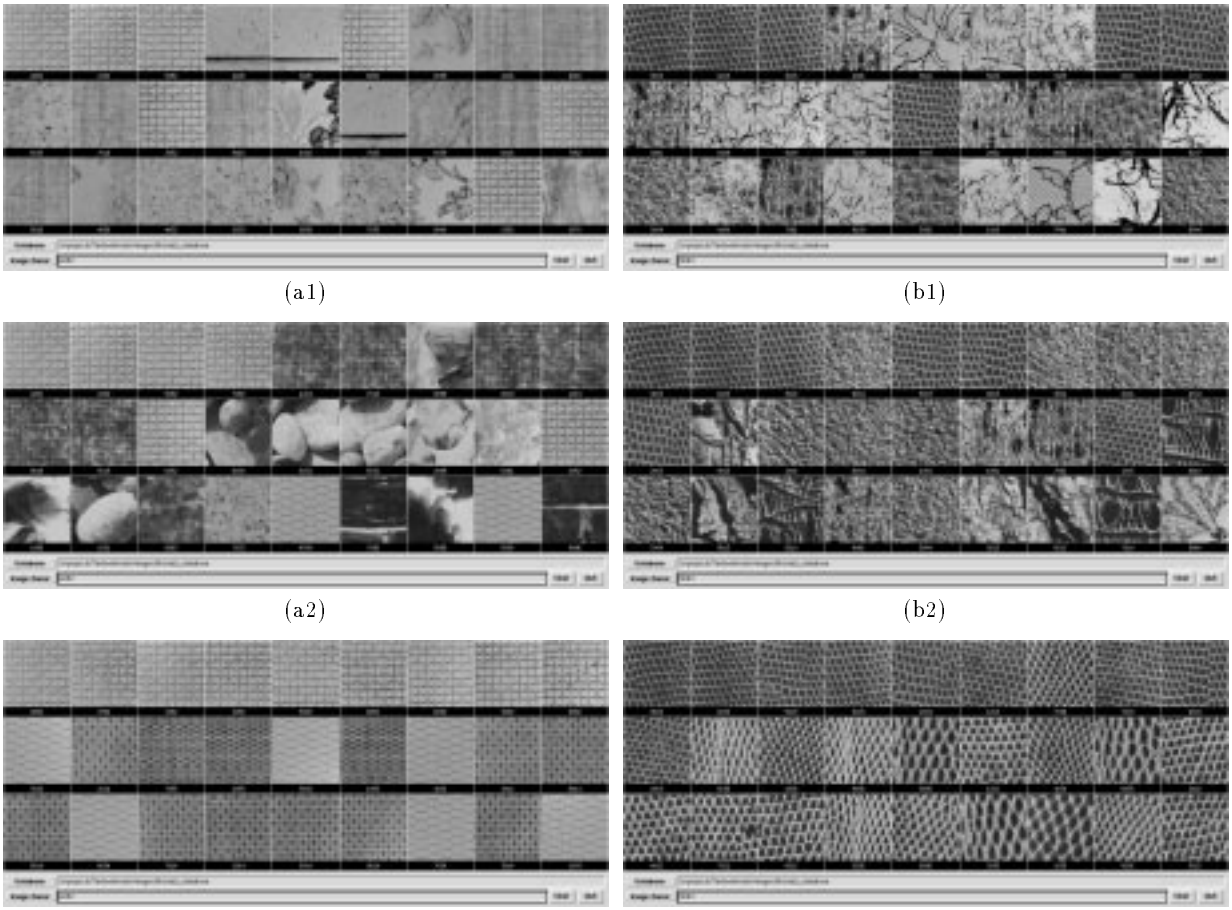
In the final stage, the two database orderings are combined using the confidence measures generated by the harmonicity test. Denote the rank of a database image in the harmonic ordering as O_h and the one in the MRSAR ordering as O_r. The joint rank of the image is computed as

$$O_{joint} = O_h P(\omega_h | r_e) + O_r P(\omega_r | r_e).$$

The final similarity ordering of the database is formed by sorting images in the ascending order of their joint rank values.

2.2.5 Image Retrieval Examples

Figure 3 shows two sets of image retrieval results of using the shift-invariant principle component analysis (SPCA) [12], the MRSAR, and the Wold-based methods over the Brodatz Database. In each picture, the upper left image is the prototype, and the retrieved images are shown by descending similarity to the prototype in raster-scan order. The Wold method demonstrates superior qualitative and quantitative performance by offering both "intra-class" accuracy and perceptually more satisfying "inter-class" similarity. The benchmarking results of five texture models, where the Wold model placed the first, can be found in [6].



(a1) (b1)

(a2) (b2)

(a3) (b3)

Figure 3: Image retrieval examples. (a1)-(a3): straw cloth pattern. (b1)-(b3): reptile skin pattern. Three methods are compared: SPCA ((a1),(b1)), MRSAR ((a2),(b2)), and Wold ((a3),(b3)). In each picture, the images are raster-scan ordered by their similarities to the image in upper left.

2.3 Natural Scene Representation

A K-means-based MRSAR feature clustering algorithm was introduced in [6] to segment natural scene images for homogeneous regions. The Wold features of these regions can be used for subsequent content identification and similarity comparison.

2.4 Perceptual Properties of Wold-based models

In the image retrieval experiment, the Wold texture model appears to offer perceptually more satisfying results. To further investigate the perceptual properties of Wold-based modeling, a psychophysical study was conducted [5].

Rao and Lohse identified the most important dimension of human texture perception as repetitiveness vs. randomness [13]. In the current study, humans and a computer program order a set of texture samples along this dimension. The correlation between the averaged human ordering and the computer ordering are used to gauge how well the computational model captures the perceptual properties of the images with respect to the perceptual axis.

2.4.1 Human Experiment

In the experiment, 32 human subjects (equal number of men and women) ordered a set of 20 Brodatz texture samples between two sets of adjectives: repetitive, non-random, directional, regular, locally oriented, and uniform vs. non-repetitive, random, non-directional, irregular, non-oriented, and non-uniform. (In Rao and Lohse's results, these adjectives label the two ends of the top perceptual axis.) The human ranking scores were then averaged and the samples re-ordered based on their average ranks to produce the final human ordering.

Frame 20 Frame 40 Frame 60 Frame 80

Figure 4: Example frames of the Walker sequence, with frame size 320×240.

2.4.2 Computer Experiment

A computer program ordered the same set of images using the Wold computational model. The orthogonal Wold components of an image have distinctive visual properties. It is conceivable that these components can be used to represent the perceptual properties of a texture pattern. Based on models of human early vision system [1][2], the total energy of the Wold components was used as the physical quantity to measure the perceptual strength of the components.

For each test sample, the computer program first performs a spectral Wold decomposition [5] to obtain the orthogonal image components. Then the signal energy of the components are computed. The ratio between the deterministic energy (including both harmonic and evanescent components) and the total energy is used for image ordering.

2.4.3 Data Analysis and Conclusions

Both Spearman and Kendall rank correlation coefficients were used to assess the correlation between the final human ranking and the computer ranking. To ensure that the ranking based on the averaged ranks is the best estimate of the "true" human ranking, the concordance of the human data was also evaluated.

The Spearman correlation coefficient for the human and computer rankings is $r_s = 0.9504$ with statistic $t = 12.96$ and significance $p < .001$, while the Kendall coefficient is $\tau = 0.7474$ with statistic $z = 4.61$ and significance $p < .001$. The Kendall concordance coefficient for the 32 sets of human ranking data is $W = 0.7874$ with statistic $\chi_r^2 = 478.72$ and significance $p < .001$. Therefore, the human and the computer rankings are significantly correlated.

The following conclusions can be drawn from the experimental results:

1. The highly significant correlation between the human and the computer texture ranking data suggests that the component energy resulting from the 2-D Wold decomposition of an image is a good computational measure for the most salient dimension of human texture perception, the dimension of repetitiveness vs. randomness.

2. The highly significant concordance of the human rankings indicates the following:

   (a) There exists a common interpretation to the semantic labels (the adjectives) associated to the perceptual dimension.

   (b) These labels indeed correspond to certain underlying criteria, upon which the human subjects agree, for texture similarity measurement.

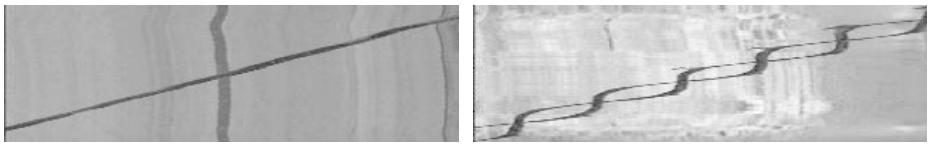
2.5 Temporal Texture Modeling for Video Analysis

2.5.1 Overview

In this work, the principle of Wold texture modeling is applied to spatiotemporal dimensions to detect, segment, and characterize periodic phenomenon in image sequences.

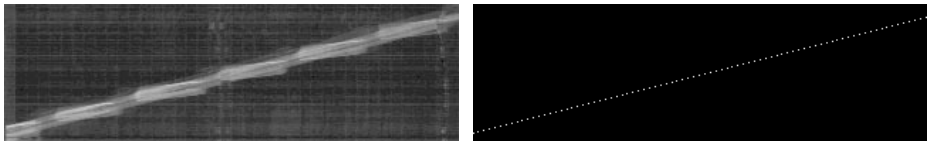
Figure 4 shows four frames of a video sequence Walker, where a person walking across the image plane. Regarding the sequence as a data cube of three-dimensions: X (horizontal), Y (vertical), and T (temporal), the XT and YT slices of the cube can reveal the temporal behavior usually hidden from the viewer. Figure 5 shows the head and ankle level XT slices of the Walker sequence. In (a), the head leaves a non-periodic straight track while the walking ankles in (b) make a crisscross periodic pattern. As it is, the periodicity in (b) is difficult to characterize.

The algorithm presented here has two stages: object tracking by frame alignment, which transforms data into a form in which periodicity can be easily detected and measured; (2) simultaneous detection and segmentation of spatiotemporal periodicity. The latter stage generates a periodicity template that not only



(a) (b)

Figure 5: Head and ankle level XT slices of Walker sequence. (a) Head level. (b) Ankle level. As it is, the periodicity in (b) is difficult to characterize.



(a) (b)

Figure 6: (a) Averaged XT image of the Walker sequence after background removal. (b) Line found in (a) by using a Hough transform method.

indicates the presence and the location of a periodic event, but also contains the fundamental frequencies and an accurate quantitative measure of how periodic the event is. Decoupling object tracking and periodicity detection conceptually modularizes the analysis process and allows the use of other tracking algorithms. In the following, the Walker sequence will be used to illustrate the key technical points. More detailed explanations of the algorithm can be found in [7].

2.5.2 Frame Alignment

In this work, a procedure is developed for the alignment of image sequences that involves little ego-motion and contains objects moving approximately frontoparallel to the camera along a straight line and at a constant speed. After frame alignment with respect to a moving object, the object should as a whole be moving in place.

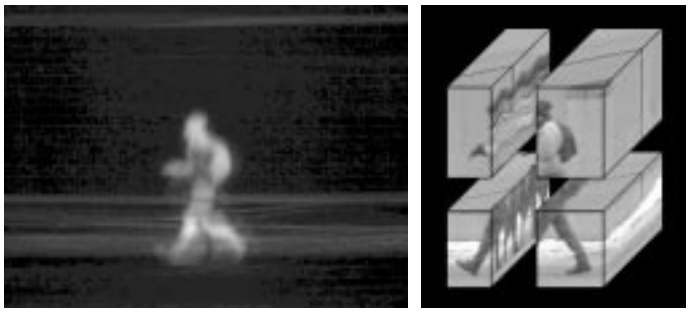
The trajectories of moving objects in the 3-D data cube is first detected. Applying 1-D median filtering along the temporal dimension, the output has mostly the still background of the sequence. The *difference sequence* between the original and the background contains mainly the moving objects. Since the object trajectories in consideration are approximately linear, the projections of the trajectories onto the XT and the YT planes (averaged XT and YT images of the difference sequence) are straight lines. These lines can be detected via a Hough transform to give the X or the Y positions of the moving objects in each frame. These position values are the *alignment indices*. The averaged XT image of the Walker difference sequence and the line found by the Hough transform method are shown in Figure 6. Each horizontal line of the pictures represents a frame, and the diagonal white line marks the object X location in each frame. Note that multiple object trajectories can be detected simultaneously using this procedure.

Using the alignment indices, image frames in a sequence are repositioned to center a moving object to any specified position in the XY plane. The aligned sequence can be cropped to save computation in subsequent processing. The location and size of the cropping window can be estimated from the average XY image of the *aligned* difference sequence. Figure 7 shows such XY image of the Walker sequence and the aligned and cropped original sequence with splits near the center of the frames to show the inside of the data cube.

2.5.3 Generating Periodicity Templates

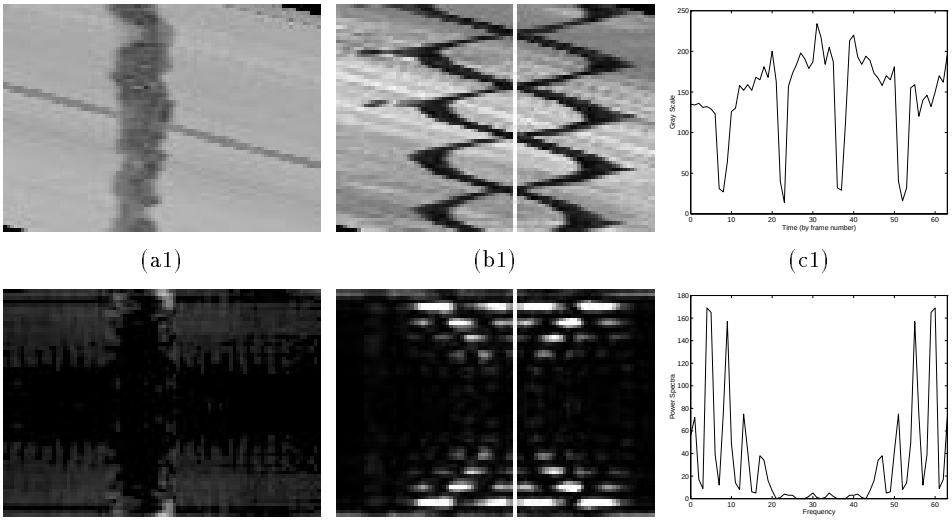
Now consider an aligned and cropped data cube. Frame pixels with the same X and Y locations form straight lines in the cube. Call these lines the *temporal lines*. Since the object of interest moves in place, its cyclic motion is reflected as re-occurring signals on some of the temporal lines. After computing the power spectrum of a temporal signal via a 1-D Fourier transform, the spectral harmonic peaks are detected and used to compute the *temporal harmonic energy* of the signal. A periodicity template is generated by using the extracted fundamental frequencies and the ratios between the harmonic energy and the total energy (the *temporal harmonic energy ratio*) along each of the temporal lines (*i.e.,* for every frame pixel locations).

Figure 8 (a1) and (b1) show the head and the ankle level XT slices of 64 frames (Frame 17 to 80) of the data cube in Figure 7 (b). Each column in the images is a temporal line. These images are the aligned and cropped version of the two XT slices in Figure 5. Columns in Figure 8 (a2) and (b2) are the 1-D power spectra of the corresponding columns in (a1) and (b1), normalized among all temporal lines in the data cube.



(a) (b)

Figure 7: (a) Averaged XY image of aligned Walker difference sequence. The area of interest is clearly shown. (b) Aligned and cropped Walker sequence with splits near the center of the frames to show the inside of the data cube.



(a1) (b1) (c1)

(a2) (b2) (c2)

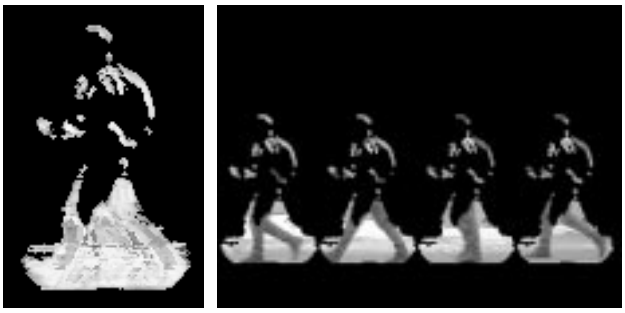
Figure 8: Signals and their power spectra along temporal lines (columns in images). (a1) and (b1): head and ankle level XT slices of aligned and cropped Walker sequence. (a2) and (b2): each column is the 1-D power spectra of the corresponding column in (a1) and (b1). (c1) and (c2): details along the white vertical lines in (b1) and (b2). Periodicity in (b1) is reflected by the spectral harmonic peaks in (b2).

Figure 8 (c1) and (c2) show details along the white vertical lines in (b1) and (b2). While the head level slice in (a1) shows no harmonicity, the periodicity of the moving ankles in (b1) is reflected by the spectral harmonic peaks in (c2).

The temporal harmonic energy ratio values of the periodicity template for the Walker sequence are shown in Figure 9 (a). The larger the energy ratio value, the more periodic energy at the location. As expected, the brightest region is the wedge shape created by the walking legs. The head, the shoulder, and the outline of the backpack are detected because the walker bounces. The hands appear at the front of the body since in most parts of the sequence the walker was fixing his gloves and moving his hands in a rather periodic manner. Note that the moving background and parts of the walker do not appear in the template since there is no periodicity present in those areas. Using the alignment indices generated at the first stage, the periodicity template can be used to mask the original sequence for the regions of periodicity in each frame. Figure 9 (b) shows the four frames in Figure 4 after they are masked and then stacked together.

The algorithm discussed above is not limited to periodicity caused by human activities. Shown in Figure 10, Wheels is a 64 frame sequence of a car passing by a building. Near the top of the building, two spinning wheels are connected by a figure 8 belt. One side of the belt is patterned and appears periodic. Every region with periodicity should be captured: the hub caps, the wheels, and one side of the belt. The algorithm accomplishes just that.

More examples can be found in [7]. Those examples demonstrate that the algorithm is well suited for detecting multiple periodicities and is robust in the presence of noise.



(a) (b)

Figure 9: (a) Temporal harmonic energy ratio values of the aligned Walker sequence. High value indicates more periodic energy at the location. (b) Using the alignment indices, the four frames in Figure 4 are masked by the template shown in (a) and then stacked together.

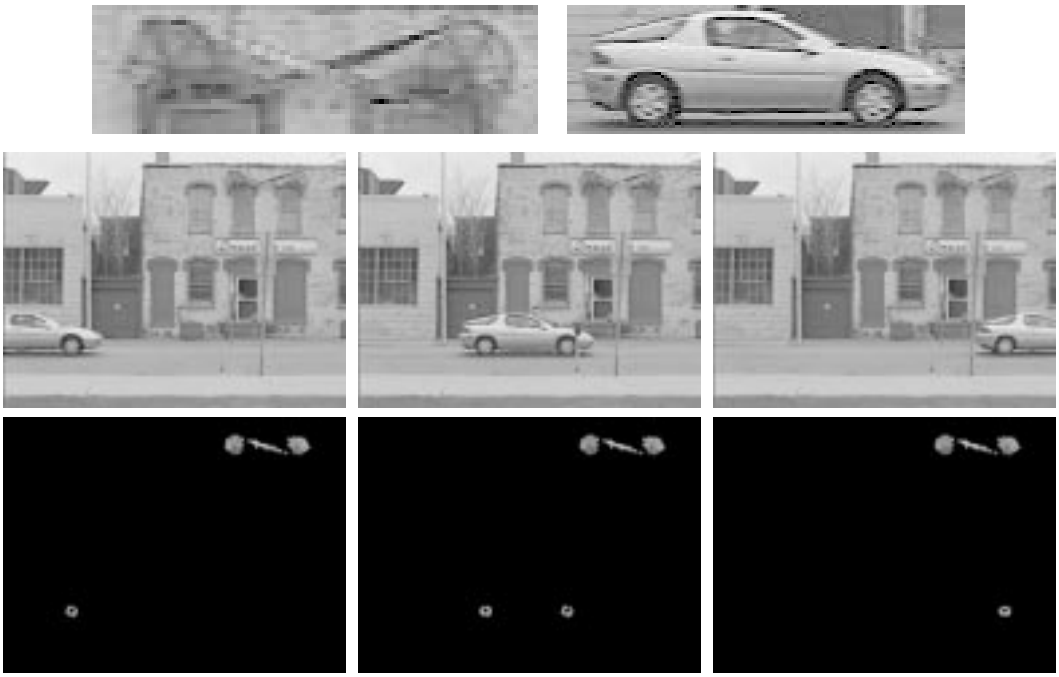


Figure 10: Wheel sequence. Top row: details of spinning wheels and car. Two bottom rows: the algorithm captures all regions with periodicity — the hub caps, the wheels, and one side of the belt.

2.6 summary

A texture model based on the 2-D Wold decomposition of homogeneous random fields is applied to image database retrieval. The Wold-based model characterizes a texture by its periodicity, directionality, and randomness, approximating what are indicated to be the three most important dimensions of human texture perception. Compared to two well-known texture models, the Wold model appears to offer a perceptually more satisfying measure of pattern similarity.

The results of a psychophysical study suggests that the component energy resulting from the 2-D Wold decomposition of an image is a good computational measure for the most salient dimension of human texture perception, the dimension of repetitiveness vs. randomness. The highly significant concordance of the human data also verifies that the top perceptual dimension found by Rao and Lohse indeed corresponds to certain underlying criteria, upon which the human subjects agree, for texture similarity measurement.

The Wold-based modeling is also applied to temporal textures. An algorithm is developed for finding periodicity in space and time. This method allows simultaneous detection, segmentation, and characterization of periodic motion in data. The resulting periodicity templates carry information on the location, frequency, and relative energy of periodic motion in a video sequence. This algorithm can also be considered as a periodicity filter, providing a model of low-level periodicity perception.

3 Scene Classification (Szummer and Picard)

Classifying images into high-level semantic classes is a very difficult task for a computer. This work [15] shows how one particular scene classification problem – classifying indoor and outdoor scenes of consumer photographs – can be approached.

Color histograms and multi-resolution autoregressive texture model coefficients are used as low-level image features. The images are also tessellated to incorporate coarse spatial position information. The classifier first classifies all the subimages. Then the full-size images are classified based on their sub-image classifications. For a collection of 518 photographs (306 outdoor and 212 indoor), this algorithm classifies 92.5% of the pictures correctly.

4 Face Detection and Recognition (Moghaddam and Pentland)

The face detection and recognition algorithm uses features resulted from the image eigenspace decomposition. The feature space consists of the eigenspace dimensions that correspond to the largest eigenvalues. For face and facial feature (eyes, nose, and mouth) detection, an unsupervised learning technique is developed [10]. This learning technique uses either a multivariate Gaussian or a mixture-of-Gaussian model to characterize the feature space. The location of a face in an image is found by using the maximum-likelihood ratio test over multi-scale. After a face is located, it is normalized to a fixed size and the facial features are detected. Using the location of the facial features, the face image is warped to align to the shape of a canonical model. Then the facial region is extracted, normalized for contrast, and projected onto a set of eigenfaces to obtain a feature vector, which is subsequently used for similarity comparison to other faces.

This face detection and recognition system placed first in the 1996 FERET (Face Recognition Technology) contest [11], which uses over 3000 images taken of people at different times and with different facial expressions.

5 Pfinder (Wren, Azarbayejani, Darrell, and Pentland)

Using a static camera, Pfinder (Person Finder) [16] is a real-time system that can find and track a person and the person's head, hands, and body while the person moves around a room.

The system uses a maximum a posteriori probability based approach. A person is modeled as a connected set of blobs (two for hands, two for feet, and one each for head, shirt, and pants). The feature set for each blob includes a pixel-level support map and the Gaussian distribution of the blob spatial location and color. The background scene is modeled as a textured surface, where every pixel is associated with a Gaussian-distributed color model. Newtonian dynamic models are used to predict the blob's position and velocity. Contour analysis is used to help initializing the blob models. For steady state tracking, the likelihood of each pixel being a member of each of the blobs and the scene is computed at each frame. The likelihood values are then used to update the support maps. Finally, all the statistical and dynamic models are updated.

The Pfinder is computationally efficient – it runs on a standard SGI Indy computer in real time. This technique has been used in applications such as gesture recognition and interactive entertainment.

6 Real-time Recognition of American Sign Language (Starner and Pentland)

American Sign Language (ASL) consists of a complex set of hand gestures. This recognition system uses Pfinder for hand tracking and hidden Markov modeling (HMM) for recognition [14].

Using one color camera the hand tracking process produces a coarse description of hand shape, orientation, and trajectory. The hand tracking data are then sent to a four-state HMM for sentence-level ASL recognition. This system interprets in real-time a forty-word subset of ASL with 99% accuracy.

7 Analysis of Discourse Video (Casey and Wachman)

This project explores ways of combining features extracted from both audio and video data for video understanding [4]. Syllabic inter-onset intervals are used for temporal segmentation. Other features include the position and velocity of the hands (use Pfinder) and the value and change in pitch. Unsupervised analysis of video is conducted via clustering in the feature space. Using this technique, a Joke Detector is built to pick out stand-up comedians Jay Leno and David Letterman's punch lines!

8 FourEyes (Minka and Picard)

Most of the existing retrieval systems require the selection of similarity measures alone with a query. This is usually a difficult task for a user. Using relevance feedback eliminates the task, and provides a more natural form of man-machine interaction.

FourEyes [9] is an extensible and self-improving interactive learning system that assists users in digital library image and video segmentation, retrieval, and annotation. The system makes tentative groupings of the data using user relevance feedback and features provided by a variety of computational models. Users no longer have to choose features or set feature control knobs. Instead, they provide positive and negative examples that allow the system to choose similarity measures automatically. FourEyes is capable of continuous learning and learns at multiple scales: on a small scale from each interaction and on a larger scale across multiple interactions.

References

[1] J. R. Bergen and E. H. Adelson. Visual texture segmentation based on energy measures. *J. Opt. Soc. of Amer. A*, 3(13), 1986.

[2] J. R. Bergen and E. H. Adelson. Early vision and texture perception. *Nature*, 333:363–364, 1988.

[3] P. Brodatz. *Textures: A Photographic Album for Artists and Designers.* Dover, New York, 1966.

[4] M.A. Casey and J.S. Wachman. Unsupervised cross-modal analysis of professional monologue discourse. In *Proc. Workshop on the Integration of Gesture in Language and Speech*, 1996.

[5] F. Liu. *Modeling Spatial and Temporal Textures.* PhD thesis, Media Arts and Sciences, MIT, Cambridge, Sept. 1997.

[6] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE T. Pat. Analy. and Machine Intel.*, 18(7):722–733, July 1996.

[7] F. Liu and R. W. Picard. Finding periodicity in space and time. In *Proc. Int. Conf. on Computer Vision*, Bombay, India, January 1998. To appear.

[8] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Patt. Rec.*, 25(2):173–188, 1992.

[9] T. Minka. An image database browser that learns from user interaction. Master's thesis, Dept. of EECS, MIT, Cambridge, MA, 1996.

[10] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. In S.K. Nayar and T. Poggio, editors, *Early Visual Learning*, pages 99–130. Oxford Univ. Press, 1996.

[11] P.J. Phillips *et al.*. The FERET September 1996 database and evaluation procedure. In *Proc. First Intl. Conf. on Audio and Video-based Biometric Person Authentication*, Crans-Montana, Switzerland, March 12-14, 1997.

[12] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. In *Proc. Int. Conf. on Acous., Speech, and Signal Proc.*, pages V–161–V–164, Minneapolis, MN, 1993.

[13] A. R. Rao and G. L. Lohse. Towards a texture naming system: identifying relevant dimensions of texture. *Vision Research*, 36(11):1649–1669, 1996.

[14] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using hidden Markov models. Perceptual Computing Section Technical Report No. 375, MIT Media Lab, Cambridge, MA, 1996.

[15] M. Szummer and R.W. Picard. Indoor/outdoor image classification. Perceptual Computing Section, MIT Media Lab, April 1997. Unpublished article.

[16] C. Wren *et al.*. Pfinder: real-time tracking of the human body. *IEEE T. Pat. Analy. and Machine Intel.*, 19(7):780–785, July 1997.

Research on Human-centered Design and Evaluation of Multimode Image Retrieval

C. Olivia Frost, School of Information
Roberta Johnson, Space Physics Research laboratory
Sang Wook Lee, Electrical Engineering and Computer Science
Yi Lu Murphy, Electrical and Computer Engineering, UM-Dearborn

Position

Image retrieval systems should be multimodal - and exploit the retrieval potential of both image and text.

Image retrieval systems should be designed for the casual user, as well as for users in research domains or with high technical skills. There is a particular need for systems which can be used by younger audiences who are growing up with much more exposure to graphic expression.

Systems for image retrieval, as with other systems designed for users, should incorporate user input in the design and development.

Project Description

We describe a project underway at the University of Michigan to explore the development and evaluation of multimode retrieval schemes which employ both image and text and are multiple path, iterative, and user-directed. The project is conducting grounded research about the creation and use of digital libraries which will support general and non-specialist users in finding information. The testbed contains digital image collections for earth and space sciences. The research is focused by the design, construction, deployment, and evaluation of a digital image library testbed for students at the middle and high school level.

The project is organized around the synergistic intersection of three sub-activities: relevant basic research in user interface design for an image library to serve middle and high school users; image classification and retrieval using text and image contents; design and construction of an evolving testbed system; and its deployment, use, and assessment. These activities are built upon the complementary strengths of a team of faculty members from information retrieval, computer engineering, and space science.

We feel that effective image retrieval systems for non-specialist users will allow users to perform searches which are multiple path, iterative, and user-directed, and part of our research focuses on the integration of text and image content search strategies. While there is a substantial amount of completed and ongoing research in both textual searching as well as content-based retrieval of images, much remains to be done to see how effectively these approaches can complement each other, and how the “casual” or nonspecialist searcher would go about using them to retrieve images.

We have developed a prototype image library currently running on the WWW (http://www.si.umich.edu/Space/). The system currently has a collection of over 1,200 images in the area of earth and space science. The system provides users three different image seeking paths: text-based browse, text-based search, and image content-based search. In designing browsing strategies in which the user’s question is framed in textual terms, we drew upon a prototype developed in an earlier project for art images. The text tag-based browse system allows the user to browse by classification categories, and the browsing path has multiple levels. Each node at the second level is associated with a set of thumbnail images. Once the user has selected a category to search, and then chooses from the items enumerated under that category, she can then choose either to see more thumbnail images or select a particular thumbnail image. The text-based search path provides a more direct textual search access, allowing the user to type in keywords within the classification categories.

Our image library also provides mechanisms for the user to retrieve images based on image content. The current image database software allows users to visually search and sort a collection of images based on intrinsic visual attributes such as color and texture and thus enables the user to search using visual information. The user selects an image and asks the system to “Give me more pictures that look like this”. The system returns the search results by displaying the images on the screen in the order of similarity to the submitted image.

We are developing a search engine which provides multiple search paths which use integrated information of text and image contents. Users can browse guided by a hierarchical classification tree and search by queries, in order to narrow the textual search to the point where the system can efficiently employ content-based analysis. We are currently augmenting the existing image retrieval software with textual retrieval strategies. Under this path, the user is guided via a classification framework from broad to progressively narrower subtopics. After the user has identified the topic areas of interest, she can then identify candidate images which can be used to perform a similarity search. This approach benefits the non-domain user who is not familiar with terminology or taxonomy or topography of the discipline, and the hierarchical structure allows the user to start out from a broad, general category and work her way down to a more specific one. If a user chooses "comet" by browsing the classification menu, she retrieves a set of thumbnail images, and can then submit the image of her choice to the image search engine to retrieve, for example, a set of comet images containing fading tails, or comets with pinwheel patterns.

In developing both manual and automatic indexing procedures, we intend to utilize methods for deriving as many image-context descriptors as possible based on the user's input and interaction. Human users can play a critical role: (1) in building image-contextual information before an image is inserted into a database, and (2) in developing semi-automatic methods for image context-based query. Our classification categories will be derived in part from our user group. In a pilot study to be implemented this summer, students and teachers from middle and high school who participate in the study will be presented with a set of images, and be asked to specify the content of each image. We will compile an initial list of classification terms based on the outcome of the user study as well as national standards in use for science education.

Our research will focus on the development of computer vision algorithms that extract high level image contents and derive similarity functions to measure the closeness of a matching between a high level image content query to an image in the image collections. For each high level image content descriptor such as "debris", "pinwheels", etc., we will present the potential users a set of primitives extracted from an image that contains the descriptor, and user's input will help us to develop a computer vision algorithm to extract the image features and match the image features with the query.

Our focus is on the usability of image retrieval systems for casual users, and on the way in which users combine image and text retrieval strategies. We are designing a retrieval system for the domain of earth and space science, and with non-specialist users in mind, specifically middle and high school students. However, our goal is also to provide a scheme which is generic and will eventually have application to many domains, including the humanities and the sciences, and to many types of users, including the general adult user.

The University of Michigan Digital Image Library (UMDIL) project will be linked to the architecture of the University of Michigan Digital Library (UMDL), sponsored by NSF/NASA/ARPA, and the Windows to the Universe project, sponsored by NASA. Windows to the Universe is an effort to develop, implement, deploy and test a World Wide Web site designed for the general public on the earth and space sciences. The UMDIL image collection testbed will be integrated into the Windows to the Universe interface, allowing immediate access to the existing audience of the project, and the outreach institutions currently working with the Windows to the Universe project (primary schools, museums, and libraries) will be involved in the evaluation and evolution of the testbed.

Finding images in large collections

David Forsyth, Jitendra Malik, Margaret Fleck, Serge Belongie and Chad Carson

U.C. Berkeley,
Berkeley,
CA 94720
USA

Abstract

Digital libraries can contain hundreds of thousands of pictures and video sequences. Typically, users of digital libraries wish to recover pictures and videos from collections based on the objects and actions depicted: this is object recognition, in a form that emphasizes large, general modelbases, where new classes of object or action can be added easily.

We first describe a representation - the ``blobworld'' representation - that uses an image segmentation in terms of novel colour and texture features to represent an image in terms of a small number of coherent regions of colour and texture. The blobworld representation allows a powerful image retrieval paradigm at the composition level in which the user is allowed to view the internal representation of the submitted image and the query results.

We then show how one can use coherent regions to recover people and animals, using a representation called a body plan. This representation is adapted to segmentation and to recognition in complex environments, and consists of an organized collection of grouping hints obtained from a combination of constraints on color and texture and constraints on geometric properties such as the structure of individual parts and the relationships between parts. Body plans are part of a more general scheme of representation for object recognition, where images are segmented into regions that have a stylised structure in shape, shading, texture or motion; objects and actions are recognised by reasoning about the spatio-temporal layout of these primitives.

We will illustrate these ideas with examples of systems running on real collections of images.

Introduction

The recent explosion in internet usage and multi-media computing has created a substantial demand for algorithms that perform content-based retrieval. The vast majority of user queries involve determining which images in a large collection depict some particular type of object. Typical current systems abstract images as collections of simple statistics on colour properties; there is much work on user interfaces that support image recovery in this abstraction. Instead, we see the problem as focussing interest on poorly understood aspects of object recognition, particularly classification and top-down flow of information to guide segmentation.

Current object recognition algorithms cannot handle queries as abstract as ``find people,'' because all are based around a search over correspondence of geometric detail, whereas typical content-based-retrieval queries require abstract classification, independent of individual variations. Existing content based retrieval systems perform poorly at finding objects, because they do not contain codings of object shape that are able to compensate for variation between different objects of the same type (e.g. a dachshund and a dalmatian), changes in posture (e.g. sitting or standing), and changes in viewpoint. Furthermore, because of the poor or absent shape representation, combinations diagnostic for particular objects cannot be learned.

Blobworld

Building satisfactory systems requires automatic segmentation of significant objects. Natural segmentations should produce regions that have coherent colour and texture. We use the Expectation-Maximization (EM) algorithm to perform automatic segmentation based on image features. EM iteratively models the joint distribution of color and texture with a mixture of Gaussians; the resulting pixel-cluster memberships provide a segmentation of the image into regions where colour and texture are coherent.

After the image is segmented into regions, a description of each region's color, texture, and spatial characteristics is produced. Regions are represented as blobs of colour and texture; an image is a composite of blobs. In a querying task, the user can access the regions directly, in order to see the segmentation of the query image and specify which aspects of

promising because blobworld captures the important elements of an image---the objects it contains---rather than simply encoding overall stuff properties.

In our system, the user composes a query by submitting an image to the segmentation/feature extraction algorithm in order to see its blobworld representation, selecting the blobs to match, and finally specifying the relative importance of the blob features. The user may also submit blobs from several different images. (For example, a query might be the disjunction of the blobs corresponding to airplanes in several images, in order to provide a query that looks for airplanes of several shades.)

We define an ``atomic query'' as one which specifies a particular blob to match (e.g., ``like-blob-1''). A ``compound query'' is defined as either an atomic query or a conjunction or disjunction of compound queries (``like-blob-1 and like-blob-2''). We might expand this definition to include negation (``not-like-blob-1'') and to allow the user to specify two blobs with a particular spatial relationship as an atomic query (``like-blob-1-left-of-blob-2''). Once a compound query is specified, we score each database image based on how closely it satisfies the compound query.

We then rank the images according to overall score and return the best matches, indicating for each image which set of blobs provided the highest score; this information will help the user refine the query. After reviewing the query results, the user may change the weighting of the blob features or may specify new blobs to match.

Blobworld is a significant improvement on colour histogram based methods, both because it allows a more detailed representation of image properties and layout, and because there is a clear path for building more complex queries.

Body plans

People and many animals can be viewed as an assembly of nearly cylindrical parts, where both the individual geometry of the parts and the relationships between parts are constrained by the geometry of the skeleton and ligaments. These observations suggest the use of a representation that emphasizes assemblies of a constrained class of primitive.

Much information is available to support segmentation and recognition: firstly, segments must be coherent, extended and have near parallel sides with an interior that appears to be hide or skin; secondly, because the 3D relationships between segments are constrained, there are relatively few assemblies of 2D segments. As a result, it is possible to tell whether a person or animal is present by determining whether there is an assembly of image segments that (a) have the right colour and texture properties and (b) form an assembly that could be a view of an acceptable configuration.

A body plan is a sequence of grouping stages, constructed to mirror the layout of body segments in people and animals. To tell whether a picture contains a person or an animal, our program attempts to construct a sequence of groups according to the body plan. For example, in the case of horses the program first collects body, neck and leg segments; it then constructs pairs that could be views of a body-neck pair, or a body-leg pair; from these pairs, it attempts to construct triples and then quadruples.

At each stage of the plan, a predicate is available which tells whether a group could correspond to some view of the segments described. For a sufficiently large collection of segments, the fact that such predicates are non-trivial follows from the existence of kinematic constraints on mammalian joints. We use a simple learning strategy for learning these predicates.

We have built two systems to demonstrate the approach. The first can very accurately tell whether an image contains a person wearing little or no clothing; the second can tell whether an image contains a horse. In each case, the approach involves pure object recognition; there is no attempt to exploit textual cues or user interaction.

Lightly clad people

The system segments human skin using colour and texture criteria, assembles extended segments, and uses a simple, hand built body plan to support geometric reasoning. A prefilter excludes from consideration images which contain insufficient skin pixels. Performance was tested using 565 target images of sparsely clad people. The system was controlled against a total of 4302 assorted control images. If images are selected on the basis of the number of skin pixels only, 448 test images are marked, but 485 control images are marked. The most selective choice of geometric test marks 241 test images and only 182 control images - almost twice as selective.

Horses

The horse system segments hide using colour and texture criteria and then assembles extended segments using a body plan to support the geometric reasoning. This body plan was learned using a bounding box classifier; the topology of the body plan was given in advance. If images are recovered on the number of hide pixels alone, 85 test images and 260

recovers 11 test images and only 4 control images. While the recall is relatively low, the selectivity is very high, meaning that the system effectively extracts image semantics.

The results are good, taking into account the abstraction of the query and the generality of the control images. The program is a practical, but not perfect, tool for extracting semantics.

The Terminological Image Retrieval Model*

Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia

Consiglio Nazionale delle Ricerche
Istituto di Elaborazione dell'Informazione
Via S. Maria 46, I-56126 Pisa, Italy,
E-mail: ⟨*lastname*⟩@iei.pi.cnr.it

Abstract. We present a model for image retrieval in which images are represented both at the *form* level, as sets of physical features of the *representing* objects, and at the content level, as sets of logical assertions about the *represented* entities as well as about facts of the subject matter that are deemed as relevant for retrieval. A uniform and powerful query language allows queries to be issued that transparently combine features pertaining to form and content. Queries are expressions of a fuzzy logical language. While that part of the query that pertains to (medium-independent) content is "directly" processed by an inferential engine, that part that pertains to (medium-dependent) form is entrusted to specialised signal processing procedures linked to the logical language by a *procedural attachment* mechanism.

1 Introduction

Due to the pervasive role of images in nowadays information systems, a vast amount of research has been carried out in the last few years on methods for retrieving images by content from large repositories. This research has produced many theoretical results, on top of which a first generation of image retrieval systems (IRSs, for short) have been built [7] and, in some cases, even turned into commercial products [2, 5]. The distinguishing feature of these systems, and of the related research prototypes, is their total disregard for a proper representation and use of *image semantics*.

This study addresses the problem of injecting semantics into image retrieval by presenting an image retrieval model in which images are represented both at the form level, as sets of physical features of the objects *representing* a slice of the world, and at the content level, as sets of properties of the real-world objects *being represented*. This model is logic-based, in the sense that the representation of image content is based on a *description logic*. Features of images pertaining to form are not represented explicitly in the description logic, as they are best dealt with outside it, i.e. by means of some digital signal processing technique. However, they impact on logical reasoning through a mechanism of "procedural

* This work has been carried out in the context of the project FERMI (n. 8134): "Formalization and Experimentation in the Retrieval of Multimedia Information", funded by the European Union under the ESPRIT Basic Research scheme.

attachments" [1], which implements the connection between (logical) reasoning about content and (non-logical) reasoning about form, thus allowing a unified query language capable of addressing both dimensions.

The resulting retrieval capability thus extends that of current IRSs with the use of semantic information processing and reasoning about image content. So far, the only attempts in this direction had been based on textual annotations to images ("captions": see e.g. [13]) or their regions, in some cases supported by the use of thesauri to semantically connect the terms occurring in the text [8]. These models permit the expression of image contents, but are weak in exploiting them, due to the well-known limitations of keyword-based text retrieval [14].

2 Representing image form

Let $\mathbb{N}$ be the set of natural numbers. A *region* is any subset of $\mathbb{N}^2$, i.e. a set of *points*. A region S is *aligned* if $min\{x \mid (x,y) \in S\} = 0$ and $min\{y \mid (x,y) \in S\} = 0$. We assume familiarity with the basic notions of digital geometry, such as neighborhood and connectedness (for details, see e.g. [11, Chapter 11]). A connected set with no "holes" is called *simply connected.*

Given a set of colours $\mathcal{C}$, a *layout* is a triple $i = \langle A^i, \pi^i, f^i \rangle$, where A^i, the *domain,* is a finite, aligned, rectangular region; π^i is a partition of A^i into non-empty connected regions $\{T_1, ..., T_n\}$, called *atomic regions;* f^i is a total function from π^i to $\mathcal{C}$, assigning a colour to each atomic region (and therefore called the *colour function*) such that no two neighbour atomic regions have the same colour; formally:

$$\forall T, T' \in \pi^i, \text{ if } T \text{ is a neighbour of } T' \text{ then } f^i(T) \neq f^i(T')$$

For notational convenience, we make explicit some of the information carried by a layout: given the layout $i = \langle A^i, \pi^i, f^i \rangle$,

- the *extended regions* π^i_e of i are defined as

  $$\pi^i_e = \{S \mid \exists T_1, ..., T_k \in \pi^i, k \geq 1,\ S = \cup_{i=1}^k T_i,\ S \text{ connected}\}$$

  The fact that we do not require S to be *simply* connected allows some interesting visual objects (e.g. the figure of a goalkeeper partly covered by an approaching ball) to be classified as extended regions;
- the *extended colour function* f^i_e of a layout i is defined as the function that assigns to each extended region S a *colour distribution* $f^i_e(S)$ (i.e. a mapping from $\mathcal{C}$ to [0,1] such that $\sum_{\{c \in \mathcal{C}\}} f^i_e(S)(c) = 1$) as follows: $\forall c \in \mathcal{C}$, $\forall S \in \pi^i_e$ such that $S = \cup_{j=1}^k T_j$ and each T_j is an atomic region:

  $$f^i_e(S)(c) = \frac{\sum_{T_j \in Z} |T_j|}{|S|}$$

  where Z is the set containing all and only the atomic regions T_j in $\{T_1, \ldots, T_k\}$ that have colour c, *i.e.* $f^i(T_j) = c$, and $|S|$ refers to the cardinality of a region S viewed as a set of points.

In general, a region S is not bound to a particular layout. This binding is realized in the notion of *grounded region*, which we define as a pair $\langle i, S\rangle$, where $i = \langle A^i, \pi^i, f^i\rangle$ is a layout and $S \in \pi^i_e$.

Let $[k]$ denote the set of the first k natural numbers. Given $m, n \in \mathbb{N}$, the *image space* $\mathbf{M}(m, n)$ is given by the set of all possible layouts of domain $[m]\times[n]$. The *image universe* $\mathcal{U} = \cup_{(i,j)\in\mathbb{N}^2}\mathbf{M}(i, j)$ is the union of all possible image spaces.

3 Representing image contents

We take the content of an image to be a *scene*, i.e. a set of possible situations indistinguishable from the visual point of view. The formalism we have chosen for representing and reasoning on image contents is a *Description Logic* (DL), namely the logic is $\mathcal{ALC}$ [12], a significant representative of the DLs family; however, our model is not tied in any way to this particular choice, and any other DL would easily fit in it. The language of $\mathcal{ALC}$ includes unary and binary predicate symbols, called *primitive concepts* (indicated by the metavariable A with optional subscripts) and *primitive roles* (metavariable R), respectively. These are the basic constituents by means of which *concepts* (i.e. "non-primitive predicate symbols") are built via *concept constructors*, according to the following syntactic rule:

$$C \longrightarrow A \mid C_1 \sqcap C_2 \mid \neg C \mid \forall R.C$$

A *crisp assertion* is an expression having one of the following forms:

- $C(a)$, where a is an *individual* and C is a concept, means that a is an instance of C; for example, `(Musician` $\sqcap$ `Teacher)(tim)` makes the individual `tim` a `Person` and a `Teacher`;
- $R(a_1, a_2)$, where a_1 and a_2 are individuals and R is a role, means that a_1 is related to a_2 by means of R (e.g. `Friend(tim,tom)`);
- $T \sqsubseteq T'$, where T and T' are both concepts or both roles, means that T is a specialization of T' (e.g. `PianoPlayer` $\sqsubseteq$ (`Musician` $\sqcap$ ($\exists$`Plays.Keyboard`))).

The first two kinds of assertions are called *simple assertions,* while any instance of the last kind is said to be an *axiom.* In order to deal with the uncertainty inherent in similarity-based retrieval, we introduce in the logic *fuzzy assertions* (see e.g. [4]), *i.e.* expressions of the form $\langle\alpha, n\rangle$ where α is a crisp assertion and $n \in [0, 1]$, meaning that α is true "to degree n". We will use the terms *fuzzy simple assertion* and *fuzzy axiom*, with the obvious meaning.

The semantics of the resulting logic relies on *fuzzy interpretations*, i.e. pairs $\mathcal{I} = (\Delta^\mathcal{I}, (\cdot)^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty set (called the *domain of discourse*) including the image universe $\mathcal{U}$, and $(\cdot)^\mathcal{I}$, the *interpretation function,* maps each concept into a function from $\Delta^\mathcal{I}$ to $[0, 1]$, and each role into a function from $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ to $[0, 1]$, so that for all $d \in \Delta^\mathcal{I}$ the following conditions are satisfied:

$$(C_1 \sqcap C_2)^{\mathcal{I}}(d) = min\{{C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\}$$
$$(\neg C)^{\mathcal{I}}(d) = 1 - C^{\mathcal{I}}(d)$$
$$(\forall R.C)^{\mathcal{I}}(d) = min_{d' \in \Delta^{\mathcal{I}}}\{max\{1 - R^{\mathcal{I}}(d, d'), C^{\mathcal{I}}(d')\}\}$$

A fuzzy interpretation $\mathcal{I}$ *is a model of* an assertion $\langle C(a), n\rangle$ ($\langle R(a_1, a_2), n\rangle$, $\langle T \sqsubseteq T', n\rangle$, respectively) iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq n$ (resp. $R^{\mathcal{I}}({a_1}^{\mathcal{I}}, {a_2}^{\mathcal{I}}) \geq n$; for all $d \in \Delta^{\mathcal{I}}$, ${T'}^{\mathcal{I}}(d) \geq n \cdot T^{\mathcal{I}}(d)$), and *is a model of* a set of fuzzy assertions iff $\mathcal{I}$ is a model of all the assertions in the set. A set of fuzzy assertions Σ *entails* a fuzzy assertion $\langle \alpha, n\rangle$ (written $\Sigma \models^f \langle \alpha, n\rangle$) iff all models of Σ are models of $\langle \alpha, n\rangle$. Given Σ and a crisp assertion β, we define the *maximal degree of truth* of β w.r.t. Σ (written $Maxdeg(\Sigma, \beta)$) to be $n \geq 0$ iff $\Sigma \models^f \langle \beta, n\rangle$ and there is no $m > n$ such that $\Sigma \models^f \langle \beta, m\rangle$.

Having settled for the tool, we now specify its use for image content representation. Let i be a layout uniquely identified, in a way to be made precise later, by the individual `i`. A *content description* δ for i is a set of fuzzy assertions, consisting of the union of four component subsets:

1. the *image identification,* a set containing only a single fuzzy assertion of the form $\langle$`Ego(i)`$, 1\rangle$, whose role is to associate, along with the layout naming function n_l (see Section 6), a content description with the layout it refers to. In particular, in what follows σ(`i`) will denote the set of the (possibly many) content descriptions whose identification is `Ego(i)`;
2. the *object anchoring,* a set of fuzzy assertions of the form $\langle$`Rep(r,o)`$, n\rangle$, where `r` is an individual that uniquely identifies a grounded region of i and `o` is an individual that identifies the object depicted by the region;
3. the *scene anchoring,* a set of fuzzy assertions of the form $\langle$`About(i,o)`$, n\rangle$, where `i` and `o` are as above. By using these assertions, an indexer can state what the whole scene shown in the image is about, and this would typically be a situation of which the image shows some salient aspect;
4. the *scene description,* a set of fuzzy simple assertions (where neither the predicates `Ego`, `Rep` and `About`, nor identifiers pertaining to layout such as the `i`'s and `r`'s above, occur), describing important facts shown in the image about the individuals identified by assertions of the previous two kinds.

While the task of components 1 to 3 is that of binding the form and content dimension of the same image, component 4 pertains to the content dimension only. Note that there may be more than one content description for the same image i; this is meant to reflect the fact that there may be multiple viewpoints under which an image may be considered.

Any of components 2 to 4 can be missing in a content description. As an example, let us consider a photograph showing a singer, Mary, performing as Zerlina in Mozart's "Don Giovanni". Part of a plausible content description for this image, named `i`, could be (for simplicity, in this example we only use crisp assertions):

{`Ego(i),About(i,o),Rep(r,mary),DonGiovanni(o),Plays(mary,zerlina)`}

4 Querying layouts

A query addressed to an image base can refer either to the form dimension, in which case we call it a *visual* query, or to the content dimension, in which case we call it a *conceptual* query. These two categories are exhaustive but not disjoint. Visual queries can be partitioned in: *concrete visual queries:* these consist of images themselves that are submitted to the system as a way to indicate a request to retrieve "similar" images; and *abstract visual queries:* these are abstractions of layouts that address specific aspects of image similarity via artificially constructed image elements and can be further categorised into:

1. *colour queries:* colour distributions that are used to retrieve images with a similar colour distribution;
2. *shape queries:* specifications of one or more shapes (closed simple curves in the 2D space) and possibly of their spatial relationships, used to retrieve images in which the specified shapes occur as contours of significant objects, in the specified relationships;

and other categories, such as spatial and texture queries [6], which will not be dealt with in this paper.

In order to query layouts, the following SPSs are introduced:

- *symbols for global matching:* in general, there will be a set of such symbols, each capturing a specific similarity criterion. Since from the conceptual viewpoint these symbols form a uniform class, we will just include one of them in our language, to be understood as the representative of the whole class. Any other symbol of the same sort can be added without altering the structure and philosophy of the language. So, for global matching we use the SPS
  - `SI(i,j)` (standing for `Similar Image`): assesses the similarity between two layouts `i` and `j`;
- *symbols for local matching:* these come in two sorts. First we have *selectors,* which are SPSs needed to select the entity to match from a layout:
  - `HAR(i,r)` (`Has Atomic Region`): a selector relating the image `i` to any of its grounded atomic regions `r`;
  - `HR(i,r)` (`Has Region`): relates `i` to any of its grounded regions `r`;
  - `HC(r,c)` (`Has Colour`): relates the grounded region `r` to its colour `c`;
  - `HS(r,s)` (`Has Shape`): relates the grounded region `r` to its shape `s`.

  Second, we have symbols for local matching, assessing similarity between local entities. Similarly for what it has been done for global matching, we include one symbol for each category of entities to be matched; so we have:
  - `SC(c,c')` (`Similar Colour`): returns the similarity between colour distributions `c` and `c'`;
  - `SS(s,t)` (`Similar Shape`): gives the similarity between shapes `s` and `t`.

The semantics of the symbols introduced so far is fixed, and is given by the functions that capture the intended meaning of each symbol, as illustrated above. For example, if $\mathcal{I}$ is any fuzzy interpretation:

$\texttt{SI}^{\mathcal{I}} : \mathcal{U} \times \mathcal{U} \rightarrow [0,1]$, assigning to each pair of layouts their degree of similarity.

A fuzzy interpretation $\mathcal{I}$ is said to be an *image interpretation* if and only if it assigns the correct semantics to the SPSs. From now on, we will use the term "interpretation" as short for "image interpretation".

5 The query language

Below, we present the *query language* of our model (*cpt* abbreviates *concept*).

$\langle \textit{image-q} \rangle ::= \langle \textit{image-cpt} \rangle \mid \langle \textit{image-q} \rangle \sqcap \langle \textit{image-q} \rangle \mid \langle \textit{image-q} \rangle \sqcup \langle \textit{image-q} \rangle$
$\langle \textit{image-cpt} \rangle ::= \exists \texttt{SI}.\{\langle \textit{layout-name} \rangle\} \mid \exists \texttt{About}.\langle \textit{content-cpt} \rangle \mid$
$\exists \texttt{HAR}.\langle \textit{region-cpt} \rangle \mid \exists \texttt{HR}.\langle \textit{bound-region-cpt} \rangle$
$\langle \textit{region-cpt} \rangle ::= \exists \texttt{HC}.\langle \textit{colour-cpt} \rangle \mid \exists \texttt{HS}.\langle \textit{shape-cpt} \rangle \mid \exists \texttt{Rep}.\langle \textit{content-cpt} \rangle \mid$
$\langle \textit{region-cpt} \rangle \sqcap \langle \textit{region-cpt} \rangle \mid \langle \textit{region-cpt} \rangle \sqcup \langle \textit{region-cpt} \rangle$
$\langle \textit{bound-region-cpt} \rangle ::= \exists \texttt{Rep}.\langle \textit{content-cpt} \rangle \mid \langle \textit{bound-region-cpt} \rangle \sqcap \langle \textit{region-cpt} \rangle \mid$
$\langle \textit{bound-region-cpt} \rangle \sqcup \langle \textit{region-cpt} \rangle$
$\langle \textit{colour-cpt} \rangle ::= \{\langle \textit{colour-name} \rangle\} \mid \exists \texttt{SC}.\{\langle \textit{colour-name} \rangle\}$
$\langle \textit{shape-name} \rangle ::= \{\langle \textit{shape-name} \rangle\} \mid \exists \texttt{SS}.\{\langle \textit{shape-name} \rangle\}$

Note that a *layout-name*, a *colour-name* and a *shape-name* are not concepts, but individuals. Queries are thus *not* concepts of $\mathcal{ALC}$, but of the DL $\mathcal{ALCO}$, which extends $\mathcal{ALC}$ with the "*singleton*" $\{\}$ operator, which given an individual `i` returns a concept $\{\texttt{i}\}$. The singleton operator is necessary in queries because it allows the reference to specific individuals. However, this added expressive power has no impact on the complexity of the image retrieval problem.

Let us reconsider the example introduced in Section 3. The images about Don Giovanni are retrieved by the query $\exists \texttt{About.DonGiovanni}$. Those showing the singer Mary are described by $\exists \texttt{HR}.\exists \texttt{Rep}.\{\texttt{mary}\}$. Turning to visual queries, the request to retrieve the images similar to a given one, named `this`, is expressed by $\exists \texttt{SI}.\{\texttt{this}\}$, and can be easily combined with any conceptual query, e.g. yielding $\exists \texttt{SI}.\{\texttt{this}\} \sqcup \exists \texttt{About.DonGiovanni}$, which would retrieve the images that are either similar to the given one or are about Don Giovanni. As far as local visual queries are concerned, the images in which there is a blue region whose contour has a shape similar to a given curve `s` are denoted by the query $\exists \texttt{HAR}.(\exists \texttt{HC}.\{\texttt{blue}\} \sqcap (\exists \texttt{HS}.\exists \texttt{SS}.\{\texttt{s}\}))$. Finally, the user interested in retrieving the images in which Mary plays Zerlina and wears a bluish dress, can use the query $\exists \texttt{HR}.\exists \texttt{Rep}.(\{\texttt{mary}\} \sqcap \exists \texttt{Plays}.\{\texttt{zerlina}\}) \sqcap (\exists \texttt{HC}.\exists \texttt{SC}.\{\texttt{blue}\})$.

6 Image bases and image retrieval

We define an *image base* as a 5-tuple $IB = \langle L, n_l, n_r, \Sigma_C, \Sigma_D \rangle$ where: (a) L is a finite set of layouts; (b) n_l is an injective *layout naming* function, mapping each

layout i in L into an individual $\mathtt{i}$, which therefore acts as a unique name for it. Note that, indirectly, n_l also associates i with the set of descriptions $\sigma(n_l(i)) = \{\delta_1, \ldots, \delta_n\}$, whose elements are the content descriptions of the image whose layout is i; (c) n_r is an injective *region naming* function, mapping each grounded region $\langle i, S\rangle$ of each layout i in L into an individual $\mathtt{r}$, which therefore acts as a unique name for it; (d) Σ_C is a finite set of content descriptions, such that each layout in L has at least one associated description (i.e. $\forall i \in L,\ |\sigma(n_l(i))| \geq 1$). "Uninterpreted" images will have a single content description containing just the image identification; (e) Σ_D is a set of fuzzy assertions representing domain knowledge.

Our image retrieval model is based on the idea that, in response to a query Q addressed to an image base $IB = \langle L, n_l, n_r, \Sigma_C, \Sigma_D\rangle$, the layout named $\mathtt{i}$ is attributed a degree of relevance n iff:

$$n = max_{\{\delta_j \in \sigma(\mathtt{i})\}}\{n_j = Maxdeg(\delta_j \cup \Sigma_D, Q(\mathtt{i}))\}$$

Let us consider an image base containing two layouts $\mathtt{i}$ and $\mathtt{j}$, such that:

$$\{\langle \mathtt{Ego(i)}, 1\rangle, \langle \mathtt{About(i,o)}, 0.8\rangle, \langle \mathtt{DonGiovanni(o)}, 1\rangle\}$$
$$\{\langle \mathtt{Ego(j)}, 1\rangle, \langle \mathtt{About(j,o)}, 0.7\rangle, \langle \mathtt{WestSideStory(o)}, 1\rangle\}$$

are in Σ_I. Moreover, Σ_C contains the following axioms:

$$\langle \mathtt{DonGiovanni} \sqsubseteq \mathtt{EuropeanOpera}, 1\rangle\ \langle \mathtt{WestSideStory} \sqsubseteq \mathtt{AmericanOpera}, 1\rangle$$
$$\langle \mathtt{EuropeanOpera} \sqsubseteq \mathtt{Opera} \sqcap (\exists \mathtt{ConductedBy.European}), 0.9\rangle$$
$$\langle \mathtt{AmericanOpera} \sqsubseteq \mathtt{Opera} \sqcap (\exists \mathtt{ConductedBy.European}), 0.8\rangle$$

Suppose we are interested in those images that are about an opera conducted by a European director. To this end, we can use the query $\exists \mathtt{About.}(\mathtt{Opera} \sqcap \exists \mathtt{ConductedBy.European})$. It can be verified that the degree of relevance attributed to $\mathtt{i}$ is 0.8, whereas that of $\mathtt{j}$ is 0.7.

We close with some implementation considerations. In order to effectively perform image retrieval as prescribed by the model defined so far, we envisage an IRS consisting of the following components: (1) a *global matching engine* for each global similarity predicate, responsible of implementing a specific kind of image global matching; to this end, each such engine will make use of the feature vectors for the layouts in the image base, stored in an apposite database, the *global matching database;* (2) a *local matching engine* for each local similarity predicate, using the feature vectors stored in *local matching databases,* of which there exists one for each considered image feature (colour, shape, etc.); (3) a *DL theorem prover,* which will handle the semantic information processing, collecting the assertions contained in the Σ_C and Σ_D components of the image base and appropriately using them in reasoning about image content; (4) a *query processor,* responsible of decomposing each query into abstract, concrete, and conceptual sub-queries, demanding the evaluation of each sub-query to the appropriate component, and then properly combining the results in order to obtain the final ranked list of images. For its operation, the query processor uses a database, called the *image structure database,* which stores the semantics

of selectors as well as naming functions. The details of these components are outside the scope of this paper. We only remark, at this point, that they are well within reach of the current technology. In particular, we have developed a theorem prover for a significant extension of the DL we use here [10].

7 Conclusions

We have presented an image data model providing a retrieval capability encompassing current similarity-based techniques and, in addition, making full and proper use of image semantics. Because the representations handled by the model have a clean semantics, further extensions to the model are possible. For instance, image retrieval by spatial similarity can be added to our model with moderate effort: at the form level, effective spatial similarity algorithms (e.g. [6]) can be embedded in the model via procedural attachment, while significant spatial relationships can be included in content descriptions by drawing from the many formalisms developed within the qualitative spatial reasoning research community [3]. Analogously, the model can be enhanced with the treatment of texture-based similarity retrieval.

References

1. F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *Proceedings of IJCAI-91, Intern. Joint Conference on Artificial Intelligence*, pages 452–457, Sydney, 1991.
2. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu. The Virage image search engine: An open framework for image management. In *Storage and Retrieval for Still Image and Video Databases IV*, volume 2670 of *SPIE Proceedings*, pages 76–87, San Jose, CA, February 1996.
3. A. G. Cohn. Calculi for qualitative spatial reasoning. In *Proceedings of AISMC-3*, Lecture Notes in Computer Science. Springer Verlag, 1996.
4. D. Dubois and H. Prade. *Fuzzy Sets and Systems*. Academic Press, 1980.
5. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, and W. Niblack. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
6. V.N. Gudivada and V.V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Trans. on Inf. Sys.*, 13(2):115–144, April 1995.
7. V. N. Gudivada and V. V. Raghavan, editors. *IEEE Computer. Special Issue on Content-Based Image Retrieval*. IEEE, Sept. 1995.
8. E. J. Guglielmo and N. C. Rowe. Natural-language retrieval of images based on descriptive captions. *ACM Trans. on Inf. Sys.*, 14(3):237–267, July 1996.
9. H. V. Jagadish, A. O. Mendelson, and T. Milo. Similarity-based queries. In *Proc. of the 14th Symp. on Principles of Database Systems*, San Jose, May 1995.
10. C. Meghini and U. Straccia. A relevance terminological logic for information retrieval. In *Proc. of SIGIR-96, the 19th ACM Int. Conf. on Research and Development in Information Retrieval*, Zurich, August 1996.
11. A. Rosenfeld and A. C. Kak. *Digital picture processing*. Academic Press, 1982.
12. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

13. A. F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proc. of SIGIR96, the 19th ACM Int. Conf. on Research and Development in Information Retrieval*, Zurich, CH, August 1996.
14. C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, GB, 1979.

This article was processed using the LaTeX macro package with LLNCS style

Browsing through and searching for similar images in astronomical archives using data density-based image icons

André Csillaghy

Institute of Astronomy, ETH-Zentrum,
CH-8092 Zurich, Switzerland
csillag@astro.phys.ethz.ch

Abstract

This article addresses the issue of retrieving images from large astonomical archives. It presents a method to define indexing features describing specific characteristics of the information contained in the image. Indexing features allow to compute a "degree of similarity" between images. In the method presented here, indexing features are derived from image icons. The latter represent symbolically the image content and are mainly used for browsing. The transition from icons to indexing features is done using a self-organizing map (SOM). In image retrieval systems, SOM-generated indexing features allow to reach high levels of retrieval precision. This is illustrated with ASPECT, a system managing the Zurich archive of solar radio spectrograms. For specific queries and for recalls less than 10%, a precision above 50% have been reached. It represents about 20% increase compared with a retrieval system based on global indexing features.

1 Overview

The efficiency and effectiveness of a retrieval system for large image archives relies on two main actions:

- *Browsing through a large number of images* allows to visualize roughly but quickly the contents of the archived images. To be quick, browsing uses lossy compressed versions of the images or symbolic image descriptions like image icons (Csillaghy, 1994).
- *Searching for similar images* allows to select the set of images to browse through. This implies the determination of some "degree of similarity" between images. The degree of similarity relies on *indexing features* that describe specific characteristics of the structures contained in the images.

Traditional methods to define indexing features rely, for instance, on text association (Murtagh, 1994), color histograms (Flickner et al., 1995), low-level image properties (Gupta and Jain, 1997) or texture description (Carson et al., 1997). To process a large number of images, the actual values of the indexing features associated with each document must be derived automatically. The automatization is problematic. Traditional methods are usually developed either for conventional photographic pictures (press photographs, museum catalogues etc.) or for small collections of images. Astronomical images archives, on the other hand, are large. Moreover, the archived images are usually noisy and mostly contain diffuse structures. Generally, the methods mentioned above cannot be applied to them.

To define indexing features for astronomical images, another approach must be used. The method presented here uses the information contained in image icons. An icon is composed of a set of *boxes* describing regions of similar texture (Section 2). Boxes are analyzed with a self-organizing map (Kohonen, 1995), which classifies the latter by shapes and volumes. This classification tells about the type of structures contained in the image (Section 3). The utilization of SOMs and icons to browse through and search for similar images is applied to the management of an archive of solar radio spectrograms (Section 4) using a system called ASPECT (ASPECT, 1996). The effectiveness of

the ASPECT system based on SOM-generated indexing features, is significantly increased compared with its effectiveness based on global indexing features (Section 5).

2 Transformation of images into icons

The method to transform image into icons have been described in details in another paper (Csillaghy, 1996). Its basic elements are recalled here.

A way to segment the image into domains with similar texture is investigated. To this end, image pixels are represented as points in a 3-dimensional attribute space, where two dimensions are given by the image axes and the third dimension is given by the color values of the parameterized pixels. The attribute space is partitioned into 3-dimensional *regions* that can hold only up to a fixed number of points. Due to this constraint, the regions have different shapes and volumes to adapt to the variable density of points in the attribute space. Consider the points in a given region. They deliver information about the local data distribution, and can be used to define a structure called a *box*. The latter represents the part of the image with similar texture.

Each individual box can be considered as a 6-tuple. Consider the r-th box, b_r, of an icon. Three values determine its position in each dimension. They are determined by the average of the points in the associated region,

$$\mu_{r,i} = \frac{1}{m_r} \sum_{k=0}^{m_r-1} x_{k,r,i} \qquad (i = 1, 2, 3), \tag{1}$$

Where m_r is the number of points in the r-th region, $x_{k,r,i}$ is the value of the k-th point in the i-th dimension and in the region r. The remaining three values determine its extension in each dimension. They are determined by the standard deviations of the points in the associated region,

$$\sigma_{r,i} = \frac{1}{m_r} \sqrt{\sum_{k=0}^{m_r-1} (x_{k,r,i} - \mu_{r,i})^2} \qquad (i = 1, 2, 3). \tag{2}$$

Hence, the box representing the r-th region is written as

$$b_r = < \mu_{r,1}, \mu_{r,2}, \mu_{r,3}, \sigma_{r,1}, \sigma_{r,2}, \sigma_{r,3} > . \tag{3}$$

The set of boxes created following the method summarized above represents an abstraction of the full image information content. For a specific application, however, only a fraction of this information is interesting. Therefore, a selection procedure is used to determine which boxes actually represent the information wanted. The selection of boxes constitutes the image icon; it is visualized by projecting the selected boxes into the plane of the image (see Figure 1).

Image icons are usually displayed in a browser, for example Netscape, as illustrated in Figure 2. They occupy only a small area on screen, thus allowing to display many of them simultaneously.

3 Derivation of indexing features from icons

3.1 General considerations

Two images are compared by determining their "degree of similarity." This implies the definition of meaningful indexing features which codify the main characteristics of the structures in the image. Indexing features can be arranged in a specific order, thus building for each image a *document description vector*. The images most similar to a given reference image are those that have their associated document description vector nearest to the document description vector of the reference image. The notion of "nearest" will be further discussed in Section 3.2.

How can indexing features for astronomical images be adequately defined? The information contained in image icons can help. Using icons as source of indexing features is attractive because the information they contain has already been selected and abstracted:

- *Selected:* The boxes building the image icons have been selected as representing interesting information; thus, the computation of the values of indexing features is not biased by irrelevant data. For example, background or disturbances, which are a significant part of the original images, do not influence the computation.

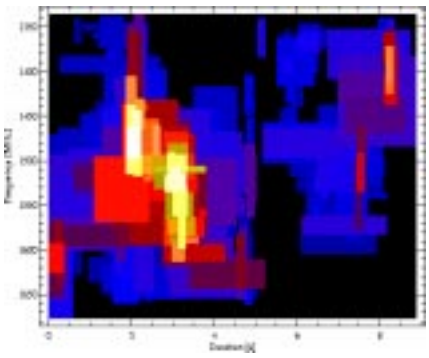


Figure 1: *Comparison between an image and its associated image icon. The image shows a solar radio spectrogram. It represents the density of the solar radio flux in the frequency-time plane. Enhanced emission is shown bright.*

– *Abstracted:* The information in icons is represented by simple 6-tuples that are easy to handle. The definition of indexing features is therefore relatively straightforward. Moreover, using icons for the computation of indexing features does not require any additional access to the large-size original documents.

Many ways can be used to define indexing features from icons. A number of *global indexing features* have been defined in another paper (Csillaghy, 1997). Furthermore, *local indexing features* can be defined using, for instance, minimal spanning trees (Murtagh and Heck, 1987), principal component analysis (Jolliffe, 1986) or self-organizing maps. The rest of the paper focuses on the utilization of the latter.

3.2 Self-organizing maps for the definition of indexing features

3.2.1 Principles of self-organizing maps

Basically, a self-organizing map (SOM) is a two-dimensional artificial neural network, that is, an array of interconnected cells. It has been described by Kohonen (1995) and is schematized in Figure 3. The spatial location of a cell in the map corresponds to a specific region of the multidimensional attribute space to be analyzed, or *input space*. A *cell* of the map reacts—that is, switches from its unactivated state 0 to its activated state 1—when a data point of the input space "presented" to the map originates from the corresponding region of the input space. The correspondence between regions of the input space and cells of the map is determined during a training process.

A set of data samples are used to train the map. During training, the map has its cells i, characterized by reference vectors m_i that are updated after each input. The learning process is described by:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)[x(t) - m_i(t)] \\ \qquad\qquad \text{if} \quad i \in N(t) \\ m_i(t) \qquad\qquad \text{otherwise,} \end{cases} \tag{4}$$

where t is the (discrete) updating time, $x(t)$ is the current input vector and $\alpha(t)$ is called the *adaptation gain*, $0 < \alpha(t) < 1$. The *neighborhood function* $N(t)$ determines which cells must be updated for a given input point. Thus, the topological structure of the input space is conserved in the map.

Once the training phase is completed, the reference vectors are left unchanged. They determine which cell must react to an (now arbitrary) input point. Since the topology of the input space is conserved by the map, nearby cells react also to nearby input classes. The conservation of the topology allows to visualize a multidimensional space in two dimensions. Therefore, maps can be used to visualize the differences between input points occuring in a data set.

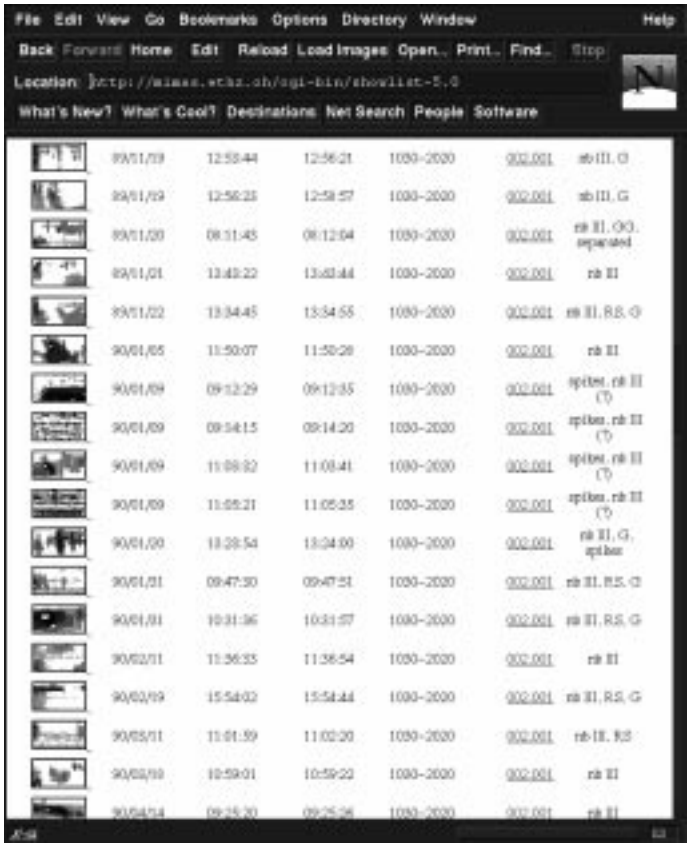


Figure 2: *This browser displays a list of images, ususally in response to a query. Here,* narrowband type III bursts (nb III) *have been requested. The information displayed consists of the following elements (from left): the (clickable) image icon, the date, the start and end time of the observation, the frequency range (in MHz), the frequency program number and remarks about the event (including the burst types).*

3.2.2 Self-organizing maps and boxes

Figure 4 illustrates how a SOM works with boxes. It is first trained to react to the shape of boxes. Then, as shown in the figure, two samples are "presented" to the trained SOM. A box of a given shape generates a cell reaction at the bottom left of the map. The other box, which has a significantly different shape, generates a cell reaction at the top right. Remarkably, if the shape of two boxes differ only slightly, their corresponding cell reaction are also nearby in the map.

The boxes contained in image icons are used as input data to SOMs. Each box produces a single cell reaction. By summing the cell reactions corresponding to individual boxes, a "total" map can be associated with a given icon. This approach is attractive for the following reasons:

- About 300 boxes are contained in a single icon. Thus, a large quantity of boxes is available in the whole archive. This allows to train the SOM with a large number of samples.
- Because a map represents a sum of cell reactions, and not only a single cell reaction, local properties of the image content can be revealed.

Indexing features are defined by associating a cell of the SOM with each single indexing feature.

3.3 Measure of the similarity between documents

Remember that indexing features are arranged in a specific order, thus building a document description vector. The distance between description vectors, in a given metric, corresponds to the similarity

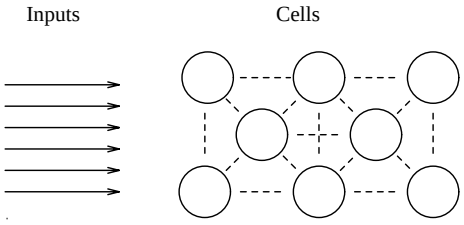


Figure 3: *A self-organizing map consists of an array of interconnected cells. A region of the input data space is associated with each cell of the network. The dimensionality of the input space is arbitrary.*

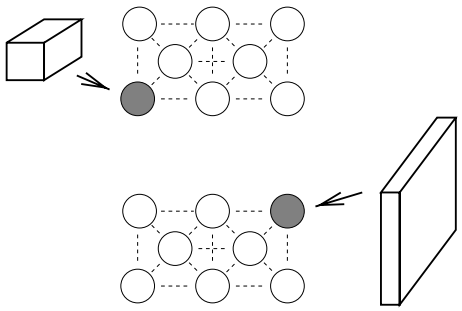


Figure 4: *A simple functional example of the self-organizing map. Assume that the map is trained to react to boxes (characterized by 6-tuples). It will organize itself so that different shapes of boxes will make different regions of the map react.*

between documents. Consider the space spanned by the description vectors, called the *description vector space*. Consider also two document description vectors, $\vec{d_j}$ and $\vec{d_k}$ in the description vector space. Their distance can be measured, for instance, using the Euclidean norm:

$$\rho_E(\vec{d_j}, \vec{d_k}) = ||\vec{d_j} - \vec{d_k}|| = \left(\sum_{i=0}^{m_\varphi - 1} (\vec{d_j} - \vec{d_k})^2 \right)^{\frac{1}{2}}. \tag{5}$$

However, ρ_E is not necessary the best measure of the "degree of similarity." It fails, for instance, if similar documents are aligned in a specific direction of the description vector space. In this case, the distance function should take into account the non-isotropic distribution of points in the description vector space.

For this purpose, another function is often used (mainly in the context of textual information retrieval), is the direction cosine:

$$\cos(\theta_{j,k}) = \frac{\vec{d_j} \cdot \vec{d_k}}{||\vec{d_j}|| \quad ||\vec{d_k}||}. \tag{6}$$

$\cos(\theta_{j,k})$ assumes a linear dependence between similar document description vectors. The case $\cos(\theta_{j,k}) = 1$ represents the highest correlation between $\vec{d_j}$ and $\vec{d_k}$, and therefore corresponds to the highest similarity.

4 Applications to solar radio spectrograms

Solar radio spectrograms consist of images, often called *events*, which display the density of the solar radio flux in the time-frequency plane. Spectrograms share the typical characteristics of astronomical images: they have a low signal-to-noise ratio, and they contain diffuse structures. The structures are divided into types and sub-types. They correspond to signatures of the emision produced by the

Main type	$\lvert\mathcal{D}^{\text{rel}}_{\text{main}}\rvert$	Subtype	$\lvert\mathcal{D}^{\text{rel}}_{\text{sub}}\rvert$
III	265	narrowband	114
		broadband	73
		large group (> 5 bursts)	26
		small group (< 5 bursts)	21
		reverse drift	31
IV	56	modulated	15
		fibers	33
		zebras	8
Blips	32	III-like	11
		patchy	21
Pulsations	58		58
Patches	39	cloudy	14
		cigar	7
		caterpillars	5
		large spots	13
Spikes	50		50

Table 1: *The classification of radio bursts in types and subtypes used for the evaluation of the ASPECT system A test collection $\mathcal{D}$ of 437 images is used. The number of images per main type and sub-type classes are given by $\lvert\mathcal{D}^{\text{rel}}_{\text{main}}\rvert$ and by $\lvert\mathcal{D}^{\text{rel}}_{\text{sub}}\rvert$, respectively.*

acceleration of high-energy particle in the solar corona. The list of types and subtypes used in this work is given in Table 1.

Parameter	Ordering phase	Fine tuning phase
Number of training samples	10,000	100,000
Radius of the neighborhood function	30	3
Adaptation factor	0.9	0.02

Table 2: *The parameters used to train the SOM.*

4.1 Parameters for training the SOM

The SOM is trained in two phases. First, the *ordering phase* determines the values of the reference vectors, to establish roughly the correspondence between the topology of the input space and the map. Second, the *fine tuning phase* adjusts accurately the values of the already ordered reference vectors.

The SOM package used in this work has been developed by the group of Kohonen (SOM, 1995). For the tests presented below, a map of a dimension of 30x30 was used. The parameters given below are determined experimentally. The are either determined specifically for ASPECT, or for SOMs in general. They are summarized in Table 2.

4.2 Maps for a single type of radio emission

Figure 5 compares maps associated with three images of the same burst type. These images belong to the class of "type III bursts." The maps present the following characteristics:

- There are two main regions where reactions are registered. The first region is around $(15, 15)$. The second region is at $(20 - 25, 5 - 10)$ and is more spreaded.
- Some regions of the map recorded no reactions at all.

– Some regions of the map recorded a relatively small number of reactions, but for each map at different locations

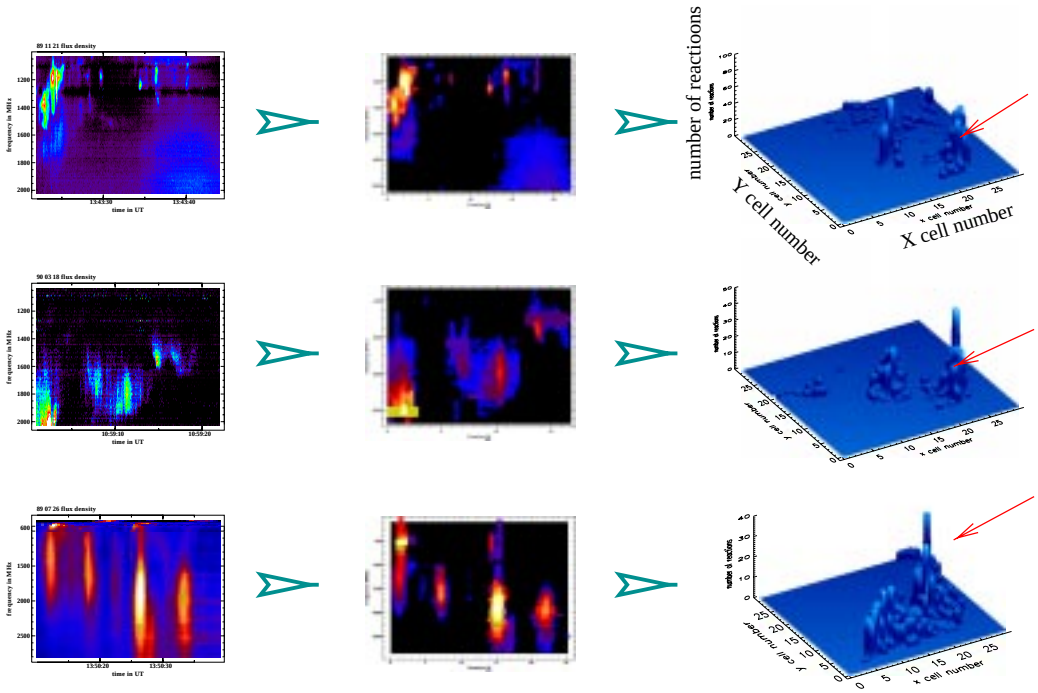


Figure 5: *Three spectrograms of the same type of radio burst. The map reactions are located in the same regions (arrows). Other regions did not react at all.*

4.3 Map for a different types of radio emisson

Figure 6 compares maps associated with three images of different burst types. These classes are: (1) type III bursts, (2) millisecond radio spikes and (3) type IV bursts. The maps present the following charateristics:

– There is no correlation between the type III-burst map and the type IV-burst map. This corresponds to expectations: type-III and type-IV bursts correspond to signatures of different processes.

– Between the type III-burst map and the millisecond-spikes map, only a few cells have reactions in common.

– Between the type IV burst-map and the millisecond-spikes map, there is a correlation in the region $(10-20, 25-30)$. For type IV however, a whole region of the map have reacted with no correlation with the other maps.

5 Retrieval effectiveness

An image retrieval system can be evaluated by considering its capacity to effectively retrieve information relevant to a user. It is called the *retrieval effectiveness*. Below, it is measured for ASPECT. The indexing features considered are derived using the method presented in the previous section.

The retrieval effectiveness is measured by the *recall* and the *precision* (van Rijsbergen, 1979). For a given query and a given number of documents retrieved, the recall gives the ratio between the

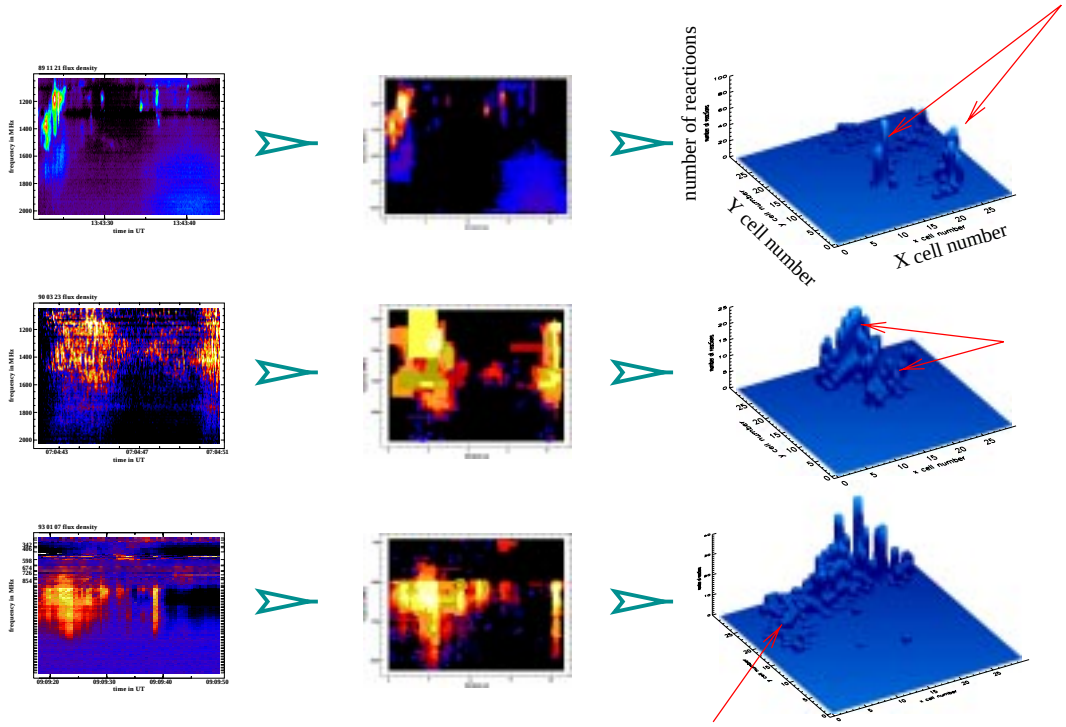


Figure 6: *Three spectrograms of different types of radio bursts. The map reactions are located in different domains (arrows). For instance there is little correlation between the first map (type III burst) and the third map (type IV burst).*

number of relevant documents retrieved and the total number of relevant documents in the collection considered. The precision gives the ratio between the number of relevant documents retrieved and the number of retrieved documents.

Recall and precision values for a system can be represented in a *recall and precision graph* (Frei et al., 1991; Raghavan et al., 1989), where the precision of the system is plotted as a function of the recall. This representation allows, for instance, to compare the effectiveness of different retrieval functions. The method to derive a recall and precision graph on the basis of these two values is described by Schäuble (1997).

The recall and precision graph for ASPECT is computed as follows. 32 reference ("query") images are selected from a test collection of 437 images. They contain solar radio bursts that are divided into several *main types* and *subtypes* (see Table 1). The classification processed by the retrieval system is compared with a classification that have been done by hand (Isliker and Benz, 1994). In this way, it is possible to decide if an image is relevant or not.

Two reference images are selected per subtype. For these images, a search for similarity is started. The SOM-based response of the system is compared with a global indexing feature-based response (Csillaghy, 1997). The similarity between the reference and the other images is computed using the Cosine function given in Equation 6. The resulting recall and precision graph is shown in Figure 7.

The graph shows that the precision is better for SOM indexing features. For low recalls, the precision is high: when considering the classes of type III bursts and type IV bursts, a precision above 50% for recalls lower than 10% is attained. Unfortunately, the precision breaks down if classes with less elements are considered. Moreover, sub-classses retrieval precision is also much lower. Nevertheless, the SOM-generated indexing features lead to a better precision than global indexing features.

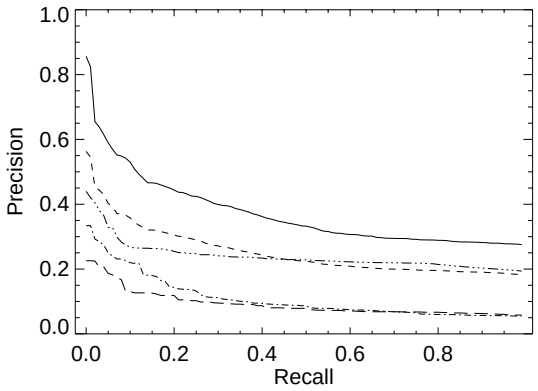


Figure 7: *The recall and precision graph for the ASPECT retrieval system. Retrieval status values are computed with the Cosine function. The following graphs are compared: Solid line: retrieval of type III and type IV bursts only. Dashed line: retrieval of all main types. Dashed-dotted line: retrieval of all subtypes. Dashed-triple-dotted line: retrieval of main types with global indexing features. Long-dashed line: retrieval of subtypes with global indexing features.*

6 Conclusions

We have shown that SOMs can be used to attain high levels of retrieval effectiveness. Methods to improve the precision, especially when considering subtypes have to be further investigated. This can be done for instance by giving more or less weight to some boxes (or to some regions of the map) when analyzing the image icons.

Acknowledgements

The author acknowledges helpful discussions with A.O.Benz, H.Hinterberger and P.Schäuble.

References

ASPECT, 1996, *The Ikarus/Phoenix Image Retrieval System ASPECT*, ETH Zurich, http://mimas.ethz.ch/archive.html

Carson, C., Belongie, S., Greenspan, H., and Malik, J., 1997, in *Proc. Workshop on content-based access of image and video libraries*, Vol. 4, IEEE

Csillaghy, A., 1994, in *Proceedings of the First Int. Conf. on Image Processing*, IEEE Computer Society Press, Los Alamitos, http://www.astro.phys.ethz.ch/papers/csillaghy/

Csillaghy, A., 1996, *Vistas in Astronomy* **40**, 503, http://www.astro.phys.ethz.ch/papers/csillaghy/nice.ps

Csillaghy, A., 1997, *Proc. of the 5th Int. Workshop on Data Analysis in Astronomy*, in press, http://www.astro.phys.ethz.ch/papers/csillaghy/

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Patrovic, D., Steele, D., and Yanker, P., September 1995, *Computer* pp 23–32

Frei, H., Meienberg, S., and Schäuble, P., 1991, in N. Fuhr (ed.), *Workshop on information retrieval*, Vol. 289 of *Informatik-Fachberichte*, pp 1–10, Springer, Berlin

Gupta, A. and Jain, R., 1997, *Comm. ACM* 40(5)

Isliker, H. and Benz, A. O., 1994, *A&AS* **104**, 145

Jolliffe, I., 1986, *Principal component analysis*, Springer

Kohonen, T., 1995, *Self-Organizing Maps*, Springer, Berlin

Murtagh, F., 1994, in *Proceedings of the 14th Int. CODATA Conference*, Chambéry, http://http.hq.eso.org/~fmurtagh/papers/image-retrieval-codata94.ps

Murtagh, F. and Heck, A., 1987, *Multivariate Data Analysis*, Astrophysics and Space Science Library, D. Reidel, Dordrecht

Raghavan, V., Jung, G., and Bollman, P., 1989, *ACM Transactions on Information Systems* **7(3)**, 205

Schäuble, P., 1997, *Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases*, Kluwer, Dordrecht

SOM, 1995, *SOM_PAK: The Self-Organizing Map Program Package*, SOM Programming team, Helsinki University of Technology, http://nucleus.hut.fi/nnrc/som_pak/

van Rijsbergen, C., 1979, *Information retrieval*, Butterworth, London

Mixing Classes and Prototypes: An Object Oriented Approach to Semantic Image modelling

Youssef Lahlou

GMD, German National Research Center for Information Technology

E-Mail: lahlou@gmd.de

Classical object oriented database management systems fail in semantic modelling of images, because such objects are not easily definable in terms of conceptual structures (classes, database schemes, ...).

Besides, object-oriented languages have evolved in various directions, including but not limited to class-based languages. Prototype-based languages represent a serious alternative to class-based ones and "organizing programs without classes" (dixit the Self language designers) is a radically different approach to object-oriented programming.

In this paper, we show that prototype-like languages can be of a great importance in semantic image modelling if the class paradigm is maintained for the sake of retrieving large collections of objects.

Semantic image modelling requires describing image structure and content within a suitable model which should provide structures and constructs capable of managing the rich variety of possible image features.

The object oriented paradigm has shown itself very suitable in most new database applications. However, adapting it to semantic image modelling is not straightforward since images are a very special case of complex objects because they can not be easily modelled through only simple use of hierarchical structures, or abstraction into class structures. Common applications more often require object content indexing rather than grouping into well-structured classes. Furthermore, unstructured objects may also refer to structured objects and vice versa, especially in multimedia environments.

Object-oriented databases make use of the class-based view of object orientation. This means that objects in such databases are structured into classes. Each object belongs to a class which abstracts its structure into a "conceptual structure" (an intensional definition of the object structure). Thus, before creating objects, a conceptual schema has to be designed to capture object structures. Concrete objects are then created with respect to the conceptual schema (they are instances of some classes).

This particular view of object-orientation is well suited for managing large collections of objects, since the conceptual schema is an invaluable source of information for querying the database, specifying and checking integrity constraints, query processing, indexing, and so on. But, in certain cases this schema is hard to design, when object structures can not be predicted before creation. This is true for image databases since semantic image content varies a lot from one image to another.

On the other hand, the prototype-based approach to object-orientation has shown itself very useful in object-oriented languages. There is no notion of class in this approach. Objects are

created either *ex-nihilo* or by "cloning" an existing object. Each object can then evolve independantly from the one it has been cloned from, especially by adding new features (attributes, methods) to its structure.

The main feature of prototype-based languages is the absence of classes. Object structures are not abstracted into some higher level conceptual entities. This feature frees the user from having to predict object structures before actually creating them.

However, when retrieving large collections of objects, the absence of classes is a major drawback, since nothing is known *a priori* about object structures and they have to be fully checked to respond to a query.

We propose a hybrid model that is able to cope with the representation and the manipulation of both unstructured and structured objects (by structured objects we mean objects for which abstract structure can be predicted within a conceptual schema). This model is particularily suited for image modelling since such objects have both common and individual features.

The model draws its inspiration from both class-based and prototype-based systems, in the sense that objects are tied to classes which only abstract a minimal structure implemented by the related objects. Each object can in turn implement an extra individual structure independantly from its class.

We thus relax the traditional instantiation link between an object and its class, and rebaptize it the *realization link*. A class is no longer a set of objects having the same structure; it is only a minimal structure that have to be implemented by each object tied to it.

Examples:
$Image : [author : Photographer,\ date : Date,\ size : 2D\_size]$ is a class.
A specific class $Monument\_image$ can be defined by augmenting the $Image$ class with the attribute $monument : Monument$.

Realization is a mechanism that links an object to a class. In other words, an object can never exist without a class to which it is tied. The mechanism of realization can be seen as the operation of giving a value to the attributes of the class by assigning an object to each of them. **The possibility is then left for objects to have in turn other additional objects in their structure, besides those of the class.** This enables objects to have individual structures, whence the improved flexibility and extensiveness of the model.

Examples:
$o_1 : [author : p_1,\ date : d_1,\ size : s_1,$ **none** $: o]$ is an object that might be a realization of class $Image$, if p_1, d_1 and s_1 are respectively realizations of $Photographer$, $Date$ and $2D\_size$ classes. An additional object named o is composing o_1 with no particular role (whence a *none* attribute).
$o_2 : [author : p_2,\ date : d_2,\ size : s_2,\ monument : m_2]$ is an object that might be a realization of class $Monument\_image$, if m_2 is a realization of class $Monument$.

The main problems arising from this model appear when it comes to querying the database. In classical structured models (relational, semantic, object-oriented), query formulation is based on class structures, assuming that objects only instanciate those structures with other objects or values.

So, when querying a database for 50 years old employees, living in Bonn, the user is aware *a priori* of the existence of a class named *Employee*, representing employees and having an attribute giving their age (say *age*) and an other giving the town they live in (say *address*). The query can then be expressed by the set:

$$\{o \in Employee, o.age = 50 \wedge o.address.town = "Bonn"\}$$

As is the case in structured data models, this kind of query can also be specified in our model, since each object realizing the *Employee* class would have an *age* attribute, and an *address* attribute, and each object realizing the class *Address* would have a *town* attribute. But, in order to fully make use of the power of the realization link, queries have to deal also with the additional part of object structures that is not in their class structures. The problem is that this structure is not known at the conceptual level; for instance, the object o_1, realizing the *Image* class uses, in its (additional individual) structure, the object o, realizing, say, the *Employee* class. This is not a mandatory (predictable) reference in the *Image* class.

Thus, the main question is how to make the user be able to use individual object components in queries so that he could specify such queries as images of 50 years old employees, living in Bonn?

In our model, a query is a quadruple $q = (c, Cl, Q, p)$ where:

- c is a class, called *target* of q,
- $Cl = \{cl_1, ..., cl_n\}$ is a set of clauses, called *criteria* of q; a clause is a logical expression based on litterals and valid paths for c (e.g. $address.town =$ "$Bonn$").
- $Q = \{q_1, ..., q_m\}$, is a set of queries, called *sub-queries* of q,
- p is a valid path for c, called *projection* of q.

Criteria, sub-queries and/or projection may be empty.

The semantics of query q is that it looks for objects realizing class c, satisfying all clause members of Cl and referencing, for each query member of Q, at least one object resulting from this query. The result of q is the set of path p destinations for all objects satisfying those three conditions.

Examples:
Query $q_1 = (c_1, Cl_1, Q_1, p_1)$ looks for 50 years old employees, living in Bonn.
$c_1 = Employee$
$Cl_1 = \{age = 50, address.town =$ "$Bonn$"$\}$
$Q_1 = \emptyset$
p_1 is empty.

Query $q_2 = (c_2, Cl_2, Q_2, p_2)$ looks for author names of images where such employees appear.
$c_2 = Image$
$Cl_2 = \emptyset$
$Q_2 = \{q_1\}$
$p_2 = author.name$.

The realization link is taken into account in our query specification language, within the set Q, which specifies criteria on all referenced objects of the main object (those predicted in the class and those proper to the object). Conceptual knowledge on objects, coming from their class structure is used in the set Cl.

To validate our approach, we designed a prototype system within the Smalltalk-80 object oriented programming environment on a SPARC station.

Semantics from Interactions in Image Databases

Simone Santini
University of California, San Diego

Abstract

Image databases have to deal with the problem of meaning. At first sight, the problem appears conceptually well defined: I want an image of a car, and I have a pretty good idea of what a car is and what it looks like. If we analyze the problem a little bit more in depth, however, we will see that assigning meaning to patches of pixels in the image is all but trivial. This is in part due to the well known technical problems of segmentation, model matching, and so on. We will argue, however, that there are much more fundamental difficulties that prevent the extraction of "objects" from the image and their automatic association with linguistic constants like "car"

As a consequence of this, the meaning of an image should be considered as a result of the interaction between the user and the database, rather than as an intrinsic property of the images. This leads naturally to replacing the "retrieval" paradigm with the "exploration" paradigm. In other words: image databases are more akin to data mining than to traditional databases.

We need to create a whole new set of concepts and tools to work in the exploration paradigm. Many of these tools are interface tools. We present some of them and discuss their utility for exploration of image databases.

Image Processing Techniques for Video Content Extraction

Inês Oliveira, Nuno Correia, Nuno Guimarães

INESC/IST, R. Alves Redol, 9, 6o, 1000 Lisboa
email: {Ines.Oliveira,Nuno.Correia,Nuno.Guimaraes}@inesc.pt

Abstract The main motivation for extracting the content of information is the accessibility problem. A problem that is even more relevant for dynamic multimedia data, which also have to be searched and retrieved. While content extraction techniques are reasonably developed for text, *video data* still is essentially opaque. Its richness and complexity suggests that there is a long way to go in extracting video features, and the implementation of more suitable and effective processing procedures is an important goal to be achieved.

This paper describes some of the basic image processing techniques offered by *videoCEL*, a toolkit for video content extraction, which makes available several commonly used abstractions and can be used by distinct applications.

Keywords

Content analysis, Video content extraction, Image processing, Temporal segmentation, Scene segmentation.

1. Introduction

The increase in the diversity and availability of electronic information led to additional processing requirements, in order to retrieve relevant and useful data: *the accessibility problem.* This problem is even more relevant for audiovisual information, where huge amounts of data have to be searched, indexed and processed. Most of the solutions for this type of problems point towards a common need: to extract relevant information features for a given content domain. A process which underlies two difficult tasks: deciding what is relevant and extracting it.

In fact, while content extraction techniques are reasonably developed for text, *video data* still is essentially opaque. Despite its obvious advantages as a communication medium, the lack of suitable processing and communication supporting platforms has delayed its introduction in a generalized way. This situation is changing and new video based applications are being developed. In our research group, we are currently developing tools for indexing video archives for later reuse, a system for content analysis of TV news [1], and hypervideo systems where hyperlinks are established based on content identification in different video streams. These applications greatly rely on efficient computational support, combining powerful image analysis and processing tools.

The developed toolkit prototype offers, in its processing components, all the functionality of these algorithms, hiding the implementation details and providing an uniform access methods to the different signal processing algorithms. The advantages offered by the use of libraries of specialised components have been largely debated [1, 4]: normalization, reutilization, flexibility, data abstraction and encapsulation, etc. The produced prototype results from the application of these principles to video content extraction, making available several abstractions commonly used by the related applications: a set of tools which extract relevant features of video data and can be reused by different applications. Next sections present a description of some of these tools and algorithms.

2. Toolkit overview

videoCEL is basically a library for video content extraction. Its components extract relevant features of video data and can be reused by different applications. The object model includes components for video data modelling and tools for processing and extracting video content, but currently the video processing is restricted to images.

At the data modelling level, the more significant concepts are the following:

- *Images*, for representing the frame data, a numerical matrix whose values can be colors, color map entries, etc.;
- *ColorMaps*, which map entries into a color space, allowing an additional indexation level;
- *ImageDisplayConvertes* and *ImageIOHandlers*, that convert images in the specific formats of the platforms and vice-versa.

Each of these concepts is represented by a (C++) class and integrated in a systematic hierarchy.

Tools for data processing are applied to the described data modelling classes, and also modelled as a hierarchy of classes: the *ImageOPs*, These operators represent functions which are applied to image regions and extract "single-image" or sequential content features. The implemented algorithms and procedures are described in more detaile in the next sections.

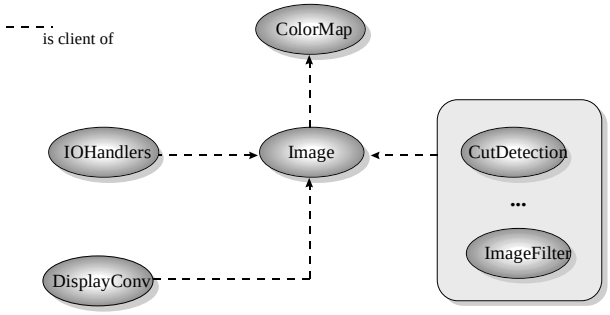


Figure 1: Object model overview.

The object model of *videoCEL* is a subset of a more complete model, which also includes concepts such has shots, shot sequences and views [1, 11]. Concepts, which are modelled in a distinct toolkit that provides functionalities for indexing, browsing and playing annotated video segments.

A shot object is a discrete sequence of images with a set of temporal attributes such as frame rate and duration and represents a video segment. A shot sequence object groups several shots using some semantic criteria. Views, are used to visualize and browse shots and shot sequences.

3. Temporal segmentation tools

One of the most important tasks for video analysis is to specify a unit set, in which the video temporal sequence may be organized [7]. The different video transitions are important for video content identification and for the definition of the semantics of the video language [8], making their detection one of the primary goals to be achieved. The basic assumption of the transition detection procedures is that the video segments are spatially and temporally continuos, and thus the boundary images must suffer significant content changes. Changes, which depend on the transition type and can be measured. The original problem is reduced to the search of suitable difference quantification metrics, whose maximums identify, with great probability, the transition temporal locations.

3.1 Cut detection

The process of detecting cuts is quite simple, mainly because the changes in content are very visible and they always occur instantaneously between consecutive frames. The implemented algorithm simply uses one of the quantification metrics, and a cut is declared when the differences are above a certain threshold. Thus, its success is greatly dependent on the metric suitability.

The results obtained by applying this procedure to some of our metrics are presented next. The thresholds selection was made empirically, while trying to maximize the success of the detection (minimizing simultaneously the false and missed detections). The captured video segment belongs to an outdoors news report, so its transitions are not very "artistic" (mainly cuts).

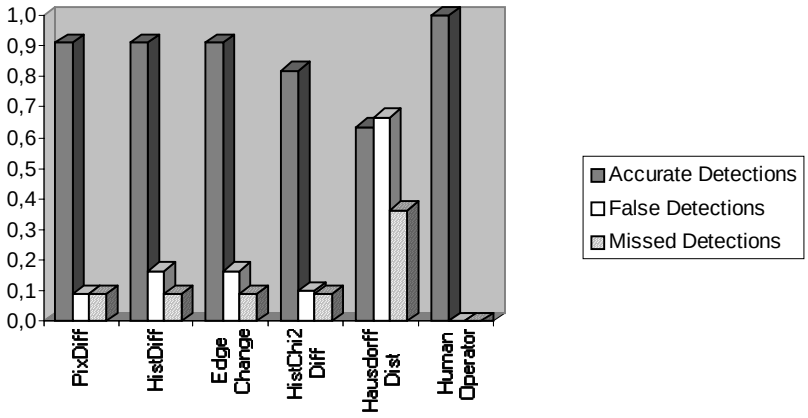


Figure 2: Cut detection results. See that almost all metrics generate a 90% accurate detection.

There are several well known strategies that usually improve this detection. For instance, the use of adaptive thresholds increases the flexibility of the thresholding, allowing the adaptation of the algorithm to diverse video content [6]. An approach that was used with some success in previous work [11], while trying to reduce some of the lacks of the metrics specific behavior, was simply to produce a weighted average of the differences obtained with two or more metrics. Pre-processing images using noise filters or lower resolution operators are also quite usual tasks, offering means for reducing image the noise and also the processing complexity. The distinctive treatment of image regions, in order to eliminate some of the more extreme values, remarkably increases the detection accuracy, specially when there are only a few objects moving on the captured scene [7].

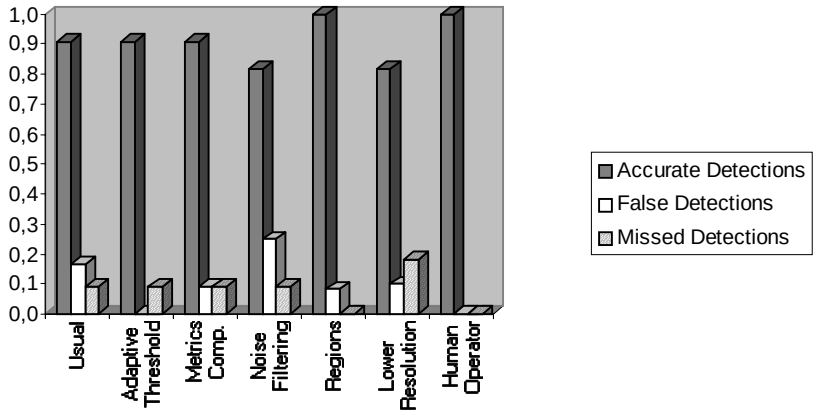


Figure 3: Cut detection results with improvements (using the *HistDiff* metric). The accuracy of the detection is clearly increased using these strategies, except in the case of noise filtering and lower resolution. One can actually explain this by defending that the images were quite clean so they were blurred with noise filtering procedure, while the use lower resolution images is essentially an approach for reducing the computation complexity.

3.2 Gradual transition detection

Gradual transitions, such as *fades*, *dissolves* and *wipes*, cause more gradual changes which evolve during several images. Although the obtained differences are less distinct from the average values, and can have similar values to the ones caused by camera operations, there are several successful procedures, which were adapted and are currently supported by the toolkit.

Twin-Comparison algorithm This algorithm [7] was developed after verifying that, in spite of the fact that the first and last transition frames are quite different, consecutive images remain very similar. Thus, as in the cuts detection, this procedure uses one of the difference metrics, but, instead of one, it has two thresholds: one higher for cuts, and another for the gradual transitions. While this algorithm just detects gradual transitions

and distinguish them from cuts, there are other approaches which also classify *fades*, *dissolves* and *wipes*, such as the Edge-Comparison presented next.

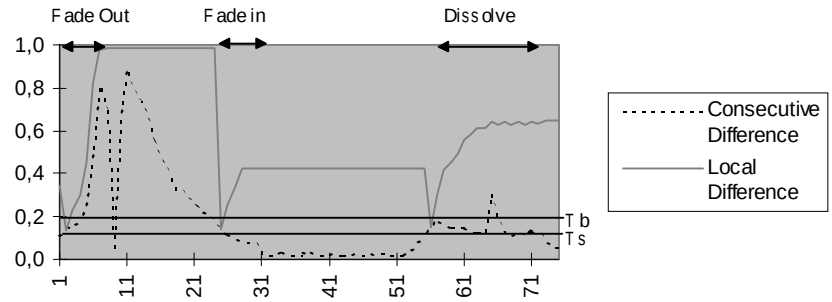


Figure 4: The *Twin-Comparison* algorithm results. When the consecutive difference is between *Tb* and *Ts*, a potential start is declared. When this happens, the local difference (the difference between the first frame of the potential segment and the current frame) starts to be computed. If consecutive frames are similar "enough" while the local difference is high, a gradual transition is declared.

Edge-Comparison algorithm This algorithm [6] analyses both edge change fractions, exiting and entering. Distinct gradual transitions generate characteristic variations of these values. For instance, a fade in always generates an increase in the entering edge fraction; conversely, a fade out causes an increase in the exiting edge fraction; a dissolve has the same effect as a fade out followed by a fade in.

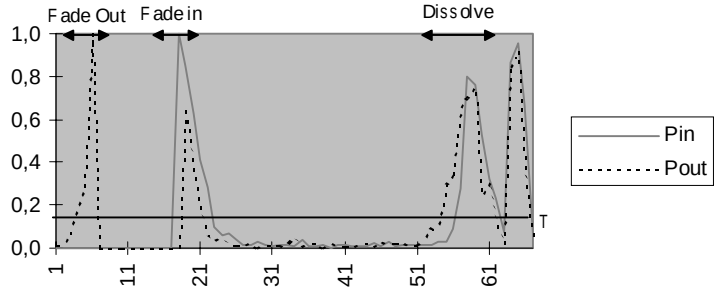


Figure 5: The Edge-Comparison algorithm results. Note that (1) in the *fade in* Pin>>Pout; (2) in the *fade out* Pout>>Pin, and (3) in the first half of the *dissolve* Pout>>Pin, and in the second half, Pin>>Pout.

4. Camera operation detection

As distinct transitions give different meanings to adjacent video segments, the possible camera operations are also relevant for content identification [8]. For example, that information can be used to build salient stills [7] and select key frames or segments for video representation. All the methods which detect and classify camera operations start from the following observation: each one generates global characteristic changes in the captured objects and background [5]. For example, when a pan happens they move horizontally in the opposite direction of the camera motion; the behavior of the tilts is similar but in the vertical axis; zooms generate convergent or divergent moves.

X-ray based method This approach [12] basically produces fingerprints of the global motion flow. After extracting the edges, each image is reduced to its horizontal and vertical projections, a column and a row, that roughly represent the horizontal and vertical global motions, which are usually referred to as the *x-ray images*.

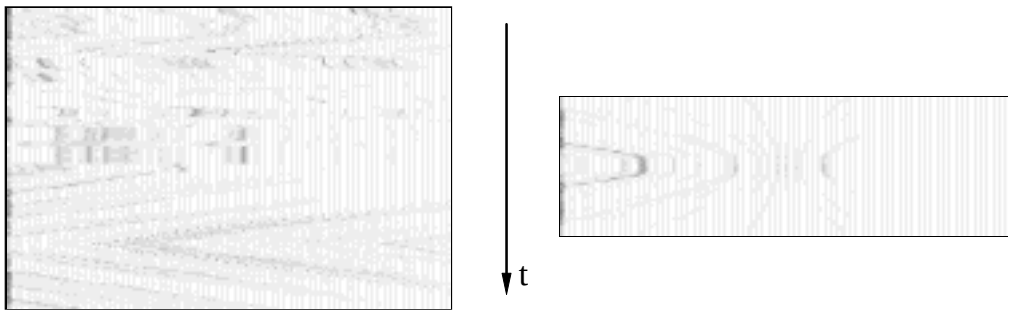


Figure 6: Horizontal x-ray images. On the left image one can see some panning operations; the right x-ray displays two zooming operations. Observing both projections it is easily perceived that (1) when the camera is still, the x-ray lines are parallel; (2) when the camera is panning or tilting, the corresponding x-ray lines slant to the opposite direction; and (3) when the lines diverge or converge, the camera is zooming.

As the above figure indicates, the behavior of the equal edge density lines, formed by the x-rays along the sequence, is characteristic of the main camera operations, giving enough information for supporting their detection. The implemented procedure basically generates the best matching percentages for each of the expected camera operations, which are then thresholded. Some of these results can be observed in the following figure, which shows all the matching percentages computed for a pan left segment.

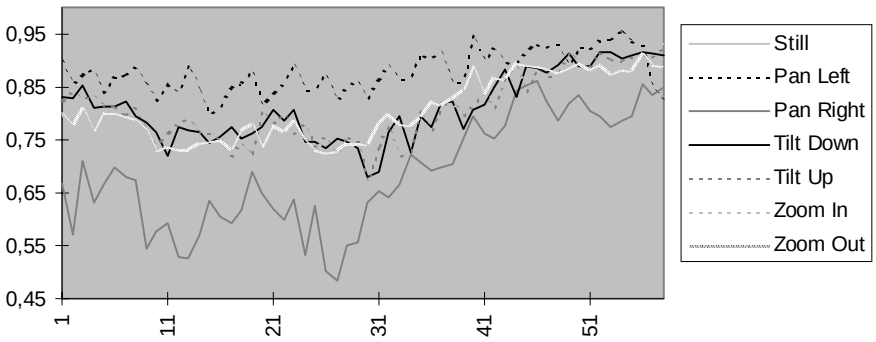


Figure 7: Pan Left results. Note that the pan left matching curve is clearly higher than the corresponding pan right results; the vertical and scaling results are also very close to each other.

As has been reported in several papers, we also intend to experiment some affine functions [6], which allow the determination of the occurred transformation between images. Although tome tests have been performed using the hausdorff distance, computed with a new multi-resolution algorithm [2], the obtained results still need further improvements.

5. Lighting conditions characterization

Light effects are usually mentioned in the cinema language grammar, as they contribution is essential for the overall video content meaning. The lighting conditions can be easily extracted by observing the distribution of the light intensity histogram: its mode, mean and average are valuable in characterising its distribution type and spread. These features also allow the quantification of the lighting variations, once the similarity of the images is determined.

Figure 8 presents some measures performed on an indoors scene, while varying its light conditions. As one will notice, the combination of these three basic measures let us easily perceive the light variations and roughly characterize the different lighting environments.

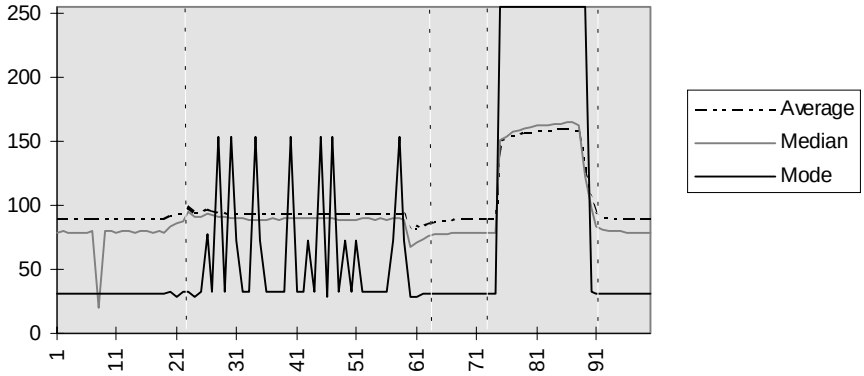


Figure 8: Luminance Statistical Measures. The first, third and fifth segments were captured in a natural light environment; the second video portion was obtain after turning on the room lights, which are fluorescent, and the fourth condition was simulated using the camera black light.

The luminance variations detection is in fact a powerful procedure, which requires further attention. There has been some trouble in distinguishing it from the changes generated by transitions. The real difficult still remains: detecting similarity when the light conditions severely change.

6. Scene segmentation

Scene segmentation refers to the image decomposition in its main components: objects, background, captions, etc. It is a first step for the identification and classification of the scene main features, and its tracking during all the sequence. The simplest implemented segmentation method is the amplitude thresholding, which is quite successful when the different regions have distinct amplitudes. It is particularly useful procedure for binarizing captions. Other methods are described below.

Region-based segmentation Region-based segmentation procedures find out various regions in an image which have similar features. One of such algorithms is the split and merge algorithm [3], that first divides the image in atomic homogeneous regions, and then merges the similar adjacent regions until they are sufficiently different. Two distinct metrics are needed: one for measuring the initial regions homogeneity (the variance, or any other difference measure), and another for quantifying the adjacent regions similarity (the average, median, mode, etc.).

Motion-based segmentation The main idea in motion-based segmentation techniques is to identify image regions with similar motion behaviors. These properties are determined by analysing the temporal evolution of the pixels. This process is carried out in the frequency image produced for all the image sequence. When more constant pixels are selected, for example, the final image is the background causing the motion removal. Once the background is extracted, the same principle can be used to extract and track motion or objects.

Figure 9: Background extraction. These images were artificially built, after determining, for each location, the sequential average, median and mode pixels values, which are shown by this order. The video sample used has about 100 frames and belongs to the initial sequence of an instructional video.

Figure 10: Object Extraction. These frames were obtained by subtracting the computed background to some image, arbitrary chosen in the sequence, that was then thresholded. The moving objects were completely extracted, specially with the median background.

Scene and object detection The process of detecting scenes or scene regions (objects) is, in certain way, the opposite process of transition detection: we want to find images regions whose differences are below a certain threshold. As a consequence this procedure uses difference quantification metrics. These functions can be determined for all the image, or a hierarchical growing resolution calculation can be performed to accelerate the process. Another tested algorithm, also hierarchical, is based in the hausdorff distance. It retrieves all the possible transformations (translation, rotation, etc.) between the edges of two images [2]. Another way of extracting objects is by representing their contours. The toolkit uses a *polygonal line approach* [3] to represent contours as a set of connected segments. The ending of a segment is detected when the relation between the current segment polygonal area and its length is beyond a certain threshold.

Caption extraction Based on an existing caption extraction method [10] a new and more effective procedure was implemented. As the captions are usually artificially added to images, the first step of this procedure is extracting high-contrast regions. This task is performed by segmenting the edge image, whose contours have been previously dilated by a certain radius. These regions are then subjected to a certain caption-characteristic size constrains, based on the x-rays (projections of edge images) properties; just the horizontal clusters remain. The resulting image is segmented and two different images are produced: one with black background for lighter text, and another with white background for darker text. The process is complete after binarizing both images and proceeding to more dimensional region constrains.

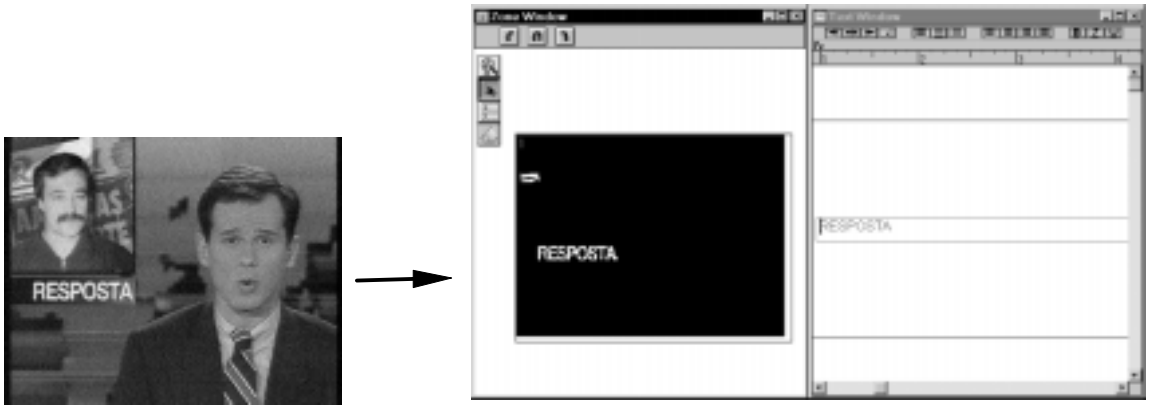


Figure 11: Caption Extraction. The right image is the result obtained after applying a commercial OCR to the frame processed by the toolkit, which is a binary image that just contains the potential caption regions.

7. Image difference quantification metrics

The accuracy of a metric is closely related to its sensitivity to changes occurred due to transitions. There are always alien factors, such as the object and camera movements, scene light changes, noise, etc., which also generates differences and may cause false detections. The metrics must be robust in these situations. The following functions were developed and tested, each one measuring the changes occurred in different features of image content:

- **Pixel differences counting** [7]**:** Counts the number of spatially correspondent pixels with different intensities, based on the principle that the transitions cause great spatial changes. It is very sensitive to

global motions and the differences introduced by the transitions are not very distinct from the average values.

- **Histogram differences sum** [7, 9]**:** Sums the differences between the histograms of both images, assuming that, unless a transition occurs, objects and background show very little color changes. These differences can be determined in several ways: χ^2, L_1, L_2, etc., with the known mathematical advantages. The pixels spatial distribution is ignored by these global measures, making them very insensitive to motion.
- **Hausdorff distance** [2]**:** Measures the maximum mismatch between two edge point sets. The edges give a preview of the image content, and are obviously affected by the transitions. This function requires high computational power and is very sensitive to noise.
- **Edge Change Rate** [6]**:** Determines the maximum of the exiting and entering edge point fractions. It is assumed that when a transition occurs, new edges appear far from the older edges, and old edges disappear far from the newer edges.

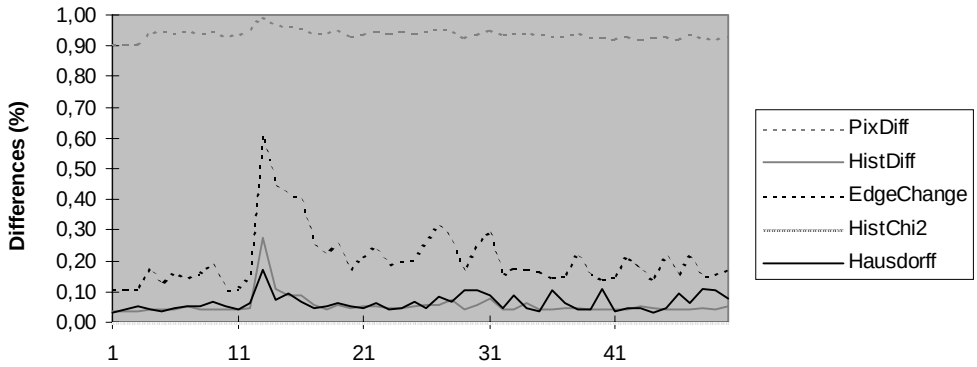


Figure 12: Differences Metrics Results. Note that all metrics have a maximum near frame 13, which clearly indicates an accentuated content change, a cut.

8. Edge detection

Two distinct procedures for edge detection [3] were implemented: (1) *gradient module thresholding*, where the image vectors are obtained using the Sobel operator; (2) the *canny filter*, considered the optimum detector, which analyses the representativity of gradient module maximums, and thus producing thinner contours. As the differential operators amplify high frequency zones, it is common practice to pre-process the images using noise filters, a functionality also supported by the toolkit in the form of several smoothing operators: the median filter, the average filter, and a gaussian filter.

9. Applications

In this section we outline the main characteristics of some applications built with the components and techniques offered in *videoCEL*.

Video browser This application [11] is used to visualise video streams. The browser can load a stream and split it in its shot segments using cut detection algorithms. Each shot is then represented in the browser main window by an icon, that is a reduced form of its first frame. The shots can be played using several view objects.

WeatherDigest The WeatherDigest application [13] generates HTML documents from TV weather forecasts. The temporal sequence of maps, presented on the TV, is mapped to a sequence of images in the HTML page. This application illustrates the importance of information models.

News analysis We developed a set of applications [1] to be used by social scientists in content analysis of TV news. The analysis was centred in filling forms including news items duration, subjects, etc., which our system attempts to automate. The system generates HTML pages with the images and CSV (Comma Separated

Values) tables suitable for use in spreadsheets such as Excel. Additionally, these HTML pages can be also used for news browsing, and there also is a Java based tool for accessing this information.

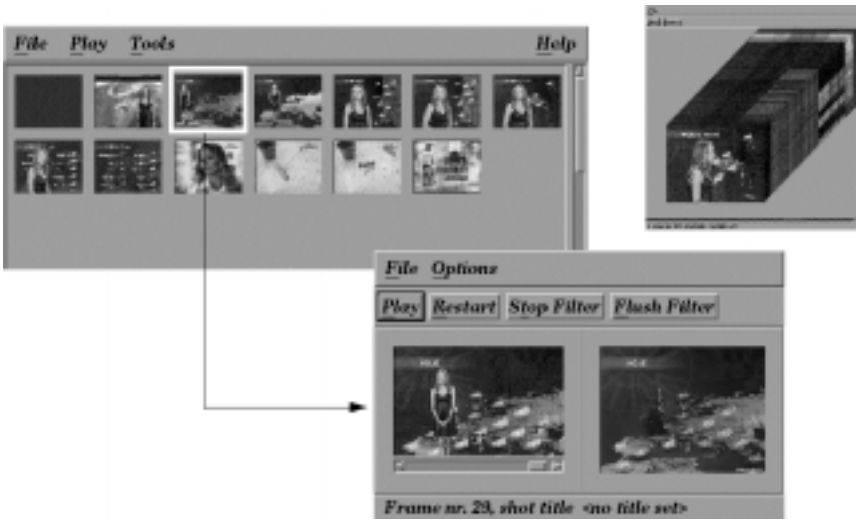


Figure 13: Video browser. The main window, and the cubic and movement filter views.

10. Conclusions and future work

The toolkit approach is a good solution when one is interested in building suitable support for extracting information content, specially because it can be reused and easily extended. While there are several efficient and normalized systems for extracting content from text and images, video related systems still remain very domain-dependent.

In this context, the components of *videoCEL* include a wide range of image processing techniques, that support the extraction of several video content features. Some of these procedures were developed specifically for video, in related works, with reported successful results. But we also have implemented several basic, but useful, image processing routines. These operations are part of the image content extraction know-how, or were simply implement to support some of the more complex operations, or the extraction of video features also considered relevant in social sciences or content analysis literature.

As future extensions, new tools will soon be added to *videoCEL* to extract additional content features. In fact, we are specially interested in including audio processing. Audio streams contain extremely valuable data, whose content is also very rich and diverse. The combination of audio content extraction tools, with image techniques, will definitely generate interesting results, and very likely improve the quality of the present analysis.

11. References

[1] Nuno Guimarães, Nuno Correia, Inês Oliveira, João Martins. “Designing Computer Support for Content Analysis: a situated use of video parsing and analysis techniques”. *Multimedia Tools and Applications Journal* (to be published during 1997).

[2] Daniel P. Huttenlocher, William J. Rucklidge. "A Multi-Resolution Technique for Comparing Images Using the Hausdorff Distance". Departement of computer Science, Cornell University, 1994.

[3] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[4] J. J. Putress, N. M. Guimarães. ”The Toolkit Approach to Hypermedia”. *Echt’90*, 1990.

[5] Y. Tonomura, A. Akutsu, Y. Taniguchi, G. Suzuki. "Structured Video Computing". NTT Human Interface Laboratories, *IEEE MultiMedia*, 1994.

[6] Ramin Zabih, Justin Miller, Kevin Mai. "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", Cornell University, 1995. (*ftp://www.cs.cornell.edu/ home/rdz/mm95.ps.gz*)

[7] Hongjiang Zhang. “Video Content Analysis and Retrieval”. *Handbook on Pattern Recognition and Computer Vision*, World Scientific Publishing Company, 1997.

[8] Arthur Asa Berger. *Media Analysis Techniques*. Sage Publications, 1991.

[9] Arun Hampapur, Ramesh Jain, Terry Weymouth. “Production Model Based Digital Video Segmentation”. *Multimedia Tools and Applications*, vol. 1, 9-46, 1995.

[10] Rainer Lienhart, Frank Stuber. “Automatic text Recognition in Digital Videos”. University of Manhein, Departement of Computer Science, *technical report TR-95-006*, 1995. (*http://www.informatik.uni.manhein.de/~lienhart/ papers/tr-95-006.gz*)

[11] Inês Oliveira, João Pedro Martins. “TV Multimedia Processing”. *Final Project Report, IST*, 1995.

[12]Y. Tonomura, A. Akutsu, K. Otsuji, T. Sadakata. “VideoMap and VideoSpaceIcon: Tools for Anatomizing Video Content”. Proceedings of *INTERCHI’93*, 1993

[13]N. Correia, I. Oliveira, J. Martins, N. Guimarães. “WeatherDigest: an experimental on media conversion”. *Integration Issues in Large Commercial Media Delivery Systems, SPIE-Photonics East’ 95, Philadelphia, USA*, vol. SPIE 2615, 50-61, 1995.

Multi-stage Classification of Images from Features and Related Text

John R. Smith
IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
jrsmith@watson.ibm.com

Shih-Fu Chang
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
sfchang@ctr.columbia.edu

Abstract

The synergy of textual and visual information in Web documents provides great opportunity for improving the image indexing and searching capabilities of Web image search engines. We explore a new approach for automatically classifying images using image features and related text. In particular, we define a multi-stage classification system which progressively restricts the perceived class of each image through applications of increasingly specialized classifiers. Furthermore, we exploit the related textual information in a novel process that automatically constructs the training data for the image classifiers. We demonstrate initial results on classifying photographs and graphics from the Web.

1 Introduction

The tremendous proliferation of visual information in the World-Wide Web is increasing the need for more sophisticated methods for automatically analyzing, interpreting and cataloging this imagery. The recent development of content-based query systems has advanced our capabilities for searching for images by color, texture and shape features [FSN+95, BFG+96, SC96]. However, these systems are limited in their capability for automatically assigning meaningful semantic labels to the images.

In this paper, we present a method for classifying images using image features and related textual information. We focus on the World-Wide Web, where a large variety of imagery consisting of graphics, animations, photographs, and so forth, is published in Web documents. The multi-stage classification system provides a hierarchy of classifiers that are trained from the images on the Web that are sufficiently annotated by text. In the successive stages, the classes are restricted as the classifiers utilize more complex features and increased training.

1.1 Related work

The classification of images in the World-Wide Web has been explored in [RF97, ASF97, FMF+96, SC97]. In [ASF97], multiple decision trees based upon image feature metrics are used for distinguishing photographs and graphics on the Web. The results are used to enhance the image search capabilities of the Webseer system. Alternatively, in order to better index the images in Web documents, Rowe and Few are developing a system for automatically associating the text in the Web documents with the corresponding images [RF97]. In [FMF+96], the images are analyzed using a blob-world representation in which objects such as people and animals are detected by matching the blobs to pre-defined body plan templates. In [SC97], as part of the WebSEEk image and video search engine, we developed a system for classifying images into subject classes using text derived from image addresses and HTML tags. We now extend this classification system to utilize image features.

1.2 Multi-stage classification system

The multi-stage image classification system consists of three stages as illustrated in Figure 1. Each stage utilizes image features and/or text. In the first stage, the images are classified into type classes, i.e., color photos, graphics, gray photos, using a decision tree based upon the analysis of image features in HSV color space. In the second stage, the images are further classified into more restricted composition classes, i.e., silhouettes, center-surround images, scenes, and textures using more complex features derived from image spatial sampling and region extraction. Finally, in the last stage, the images are classified into

semantic classes, i.e., beaches, buildings, nature, sunsets, and so forth, using specialized classifiers which are trained from images that are classified from their related text.

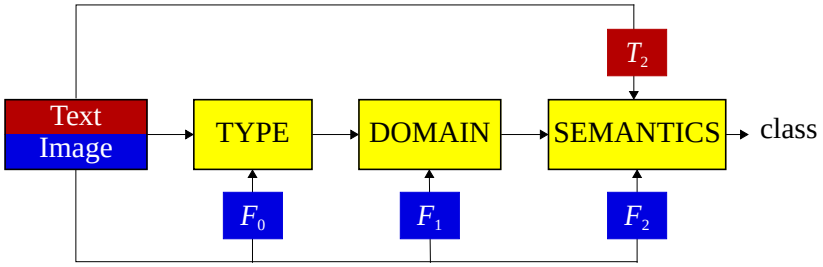


Figure 1: Multistage image classification system uses image feature sets: $\mathcal{F}_0$, $\mathcal{F}_1$, $\mathcal{F}_2$, and related text $\mathcal{T}_2$.

In this paper, we present the multi-stage image classification system and describe the processes for classifying the images into the type, composition and semantic classes. In Section 2, we introduce a new simple feature decision tree for determining image type. We present, in Section 3, the image composition classification system. Finally, in Section 4, we present a novel semantics classification system, which uses composite region templates (CRTs). We evaluate the performance of the CRT-based semantics classification system in classifying images from eight semantics classes.

2 Stage 1 – image type

In the first stage, images are classified into image type classes. The image type hierarchy is illustrated in Figure 2. We define the following set of image type classes: color photographs, complex color graphics, simple color graphics, gray photos, gray graphics, b/w (bi-level) photographs, and b/w graphics. The type classes are given by the root nodes of the decision tree in Figure 2.

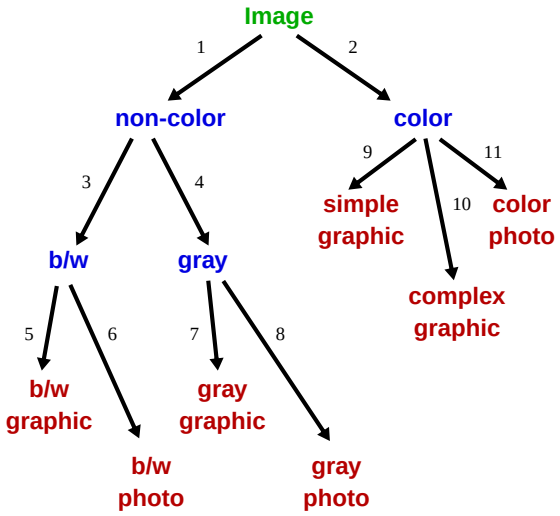


Figure 2: Image type hierarchy with five decision points: $(1, 2)$, $(3, 4)$, $(5, 6)$, $(7, 8)$, $(9, 10, 11)$.

2.1 Image type features

In order to automatically classify the images into type classes, the system analyzes the image features in HSV color space. The transformation and quantization to 166 HSV colors is given in [Smi97]. The following HSV color features are extracted from the images:

- A = relative amount of black,
- B = relative amount of white,

- C = relative amount of gray,
- D = relative amount of colors which are fully saturated, i.e., saturation = 1,
- E = relative amount of colors which are half saturated, i.e., saturation ≥ 0.5 and saturation < 1,
- F = number of colors present from the 166-color quantized HSV color space,
- G = number of grays present from the 166-color quantized HSV color space,
- H = number of hues present from the 166-color quantized HSV color space,
- I = number saturations present from the 166-color quantized HSV color space.

Table 1 gives the average feature values for the image type classes obtained from a training set of several thousand images retrieved from the World-Wide Web.

Image type	A black	B white	C gray	D fully sat.	E half sat.	F # colors	G # grays	H # hues	I # sats.
color photo	0.18	0.06	0.14	0.04	0.10	11.9	111	54	94
complex graphic	0.07	0.03	0.06	0.18	0.23	29.8	77	76	80
simple graphic	0.16	0.26	0.18	0.16	0.07	3.8	17.8	8.0	14.4
gray photo	0.24	0.06	0.70	0	0	0	130	1	1
gray graphic	0.21	0.29	0.49	0	0	0	23.2	1	1
b/w photo	0.60	0.40	0	0	0	0	2	1	1
b/w graphic	0.41	0.59	0	0	0	0	2	1	1

Table 1: Image type classes and corresponding attributes obtained from training images.

Starting at the root node in the decision tree, images are classified into increasingly specific type classes. In order to derive the decision criteria, we computed the image color features for a large set of training images. For each decision point, we identified the subset of features that were relevant to that decision. For example, for decision point $(1, 2)$, image features A, B and C are sufficient.

For each decision point, a multi-dimensional space was generated, such that each dimension corresponds to one of the relevant features (i.e., A, B, C). This multi-dimensional space was then partitioned adaptively to the training images. The frequencies by which training images of each type occur within the partitions determines the decision criteria. In this way, a new image is quickly classified by simply obtaining the most likely class in the partition corresponding to its feature values.

2.2 Adaptive partitioning

The M dimensional decision space is iteratively partitioned as follows, where τ is a training threshold ($\tau = 0.9$):

1. Assign training images to points in the M dimensional decision space by measuring their feature values.
2. Assign initial partition R_0 to the entire M dimensional decision space.
3. Split R_0 into 2^M partitions by bi-secting R_0 along each dimension.
4. For each new partition R_j, if $\neg\exists C_k$ such that $P(C_k|R_j) > \tau$ then split R_j, and repeat Step 3 and 4 as necessary.
5. For each partition R_l after all splitting, assign the likelihood of each class C_k to each partition R_l as follows:

$$P(C_k|R_l) = \frac{P(R_l|C_k)P(C_k)}{P(R_l)},$$

where $P(R_l|C_k)$ is the number of training points in partition R_l that belong to class C_k, $P(C_k)$ is the number of training points from class C_k, and $P(R_l)$ is the number of points in partition R_l.

2.3 Type classification

Given the partitioned decision space, the type class of an unknown image is determined by simply looking up which class C_k maximizes $P(C_k|R_l)$, where R_l is the partition corresponding to the features of the unknown image.

3 Stage 2 – image composition

In the second stage, the images are assigned to one of the following composition classes: silhouettes, center-surround images, scenes and textures. The image composition is determined by the separation of the center and surround areas in the image.

3.1 Center-surround separation

The image center and surround are separated by using two methods of sampling the surround areas of the image, depicted in Figure 3 as regions 'A,' 'B,' 'C,' and 'D.'

1. Method 1: most prominent color – From regions A, B, C, D, the most prominent color in the surround, i.e., given by m, where, $\forall m \neq k, h_S[m] \geq h_S[k]$, is back-projected onto the image (see [Smi97] for details about back-projection) to extract the surround region, depicted in Figure 3 as S_1.

2. Method 2: pooled color histogram – From regions A, B, C, D, a pooled color histogram is generated as follows: $\mathbf{h}_S = \mathbf{h}_A + \mathbf{h}_B + \mathbf{h}_C + \mathbf{h}_D$. Then $\mathbf{h}_S$ is back-projected onto the image to more completely extract the surround region, depicted in Figure 3 as S_2.

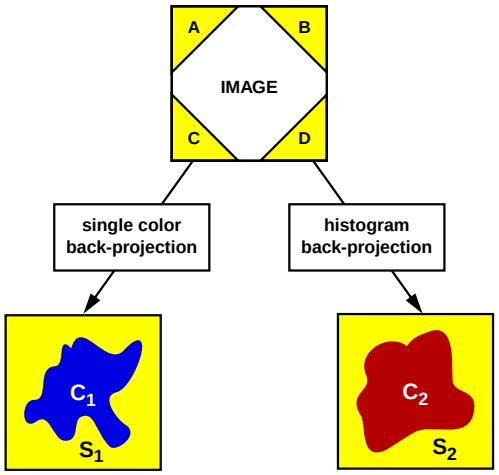


Figure 3: Image center-surround separation process for image composition classification extracts two versions of the center regions (C_1, C_2) and surround regions (S_1, S_2).

Method 1 (back-projecting the most prominent surround color) is more suited for extracting a silhouetted object that is depicted on a single color background. Method 2 (back-projecting the pooled surround histogram $\mathbf{h}_S$) is more suited for separating a multi-color surround from a center region. The results of the back-projections yield two versions of a center object, denoted by C_1 and C_2. The attributes of the extracted center regions (C_1, C_2) and surround regions (S_1, S_2) are used to determine the image composition class.

The attributes used for image composition classification are derived from the sizes of C_1 and C_2, and the color distances between C_1 and S_1, and C_2 and S_2, respectively. Table 2 indicates the typical values of the image features used for composition classification. The 'size' features indicate the relative sizes of the extracted image center regions. The 'dist' features indicate the distances in HSV color space between the respective center and surround regions.

Figure 4 illustrates the results from the center-surround separation process for the four image composition classes. For the silhouette images, Methods 1 and 2 produce similar results since the surround

Image composition	size(C_1)	size(C_2)	dist(C_1, S_1)	dist(C_2, S_2)
silhouette	0.59	0.58	0.89	0.68
center-surround	0.54	0.23	0.69	0.54
scene	0.83	0.19	0.40	0.23
texture	0.14	0.05	0.19	0.12

Table 2: Image composition classes and corresponding center-surround features.

typically contains a single color. For the center-surround images, Method 2 extracts a larger surround than Method 1 since the surround contains more than one color. Furthermore, the color distance between the center and surrounds in both cases is relatively large. In the case of the scene images, Method 2 extracts a large surround region while method 1 extracts a small surround region. Finally, for textures, both methods fail at separating a center from the surround.

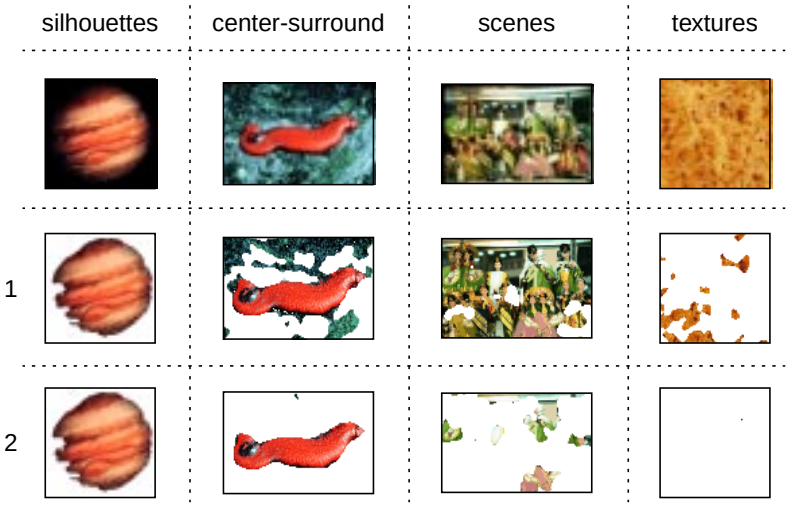


Figure 4: Center-surround separation examples using Methods 1 and 2 for the four image composition classes.

3.2 Composition classification

Given the image composition feature set, the decision space is derived from training images using adaptive partitioning of the 4-dimensional feature space. The classification of an unknown image is performed by simply extracting the center-surround features and finding the partition corresponding to the feature values. Similar to the case for image type classification, the composition label is assigned by the most likely composition class in the partition.

4 Stage 3 – image semantics

In the final stage, the images are classified into semantics classes derived from a semantics ontology (described in [SC97]). Here, we examine eight semantics classes: beaches, buildings, crabs, divers, horses, nature, sunsets, and tigers.

4.1 Text-to-subject mapping

The semantics classes are defined by identifying training images on the Web that are associated with relevant text. These images are assigned to the semantics classes by mapping the key-terms to semantics classes[1]. For example, the key-term 'sunset' is mapped into semantics class 'nature/sunsets.' This process

[1] The WebSEEk demo: `http://disney.ctr.columbia.edu/webseek`

is described in more detail in [SC97]. We now describe how the images that cannot be semantically classified using text due to lack of useful annotations, are classified using images features based upon composite region templates.

4.2 Composite region templates

The composite region templates (CRTs) are defined from training images from the semantic classes. The system extracts the color regions from the images and generates a set of region strings for each semantic class. The region strings for each class are then consolidated into the sets of CRTs.

4.2.1 Region string generation

The region strings are generated in a series of five vertical scans of the image which order the extracted regions from top-to-bottom. The five vertical scans are equally spaced horizontally. Since the images are normalized to 100 pixels, each vertical scan covers a 20-pixel wide area. In each scan, the symbol value of each consecutive region is concatenated onto the scan's region string. In general, the symbol values (i.e., symbol 'A,' 'B,' 'C' in Figure 5) represent the index values of the features of the regions.

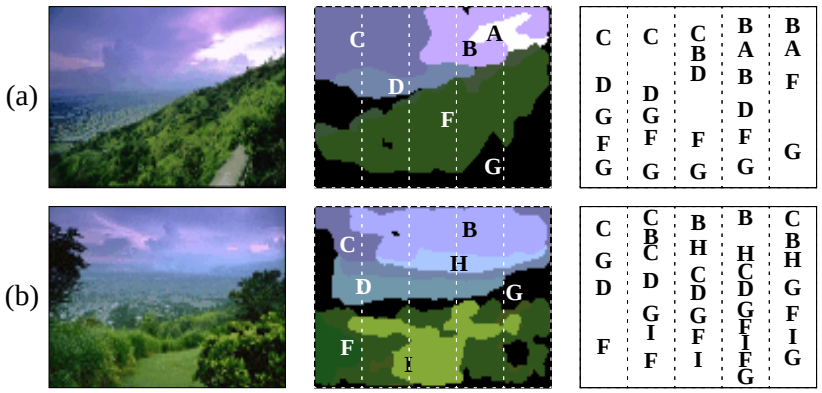


Figure 5: Examples of region extraction and region string generation using a top-to-bottom orderings (a) (CDGFG, CDGFG, CBDFG, BABDFG, BAFG), (b) (CGDF, CBCDGIF, BHCDGFI, BHCDGFIFG, CBHGFIG).

An example of the region string generation process for two nature images is illustrated in Figure 5. We can see that for the two nature images, the symbols 'A,' 'B,' and 'C' (sky) typically preceed symbols 'F,' and 'G' (grass). The objective of the CRT method is to detect these important relationships between regions for each semantic class. The top-to-bottom scans capture the relative vertical placement of the regions. Note that the five region strings from an image are not subsequently distinguished by the horizontal position of the scan.

Definition 1 Region String. *A region string* $\mathbf{S}$ *is a series of symbols* $\mathbf{S} = s_0 s_1 s_2 \ldots s_{N-1}$*, which is generated from the regions of an image where* s_n *is the symbol value (i.e., color index value) of the* n^{th} *successive region in a top-to-bottom scan.*

4.2.2 Region string consolidation

After the region strings are generated, they are consolidated to generate the CRTs in order to capture the recurring arrangements of the regions within the images and semantic classes. The CRTs characterize, in general, the order of the symbols in the region strings but not their adjacency. The likelihood of these CRTs within and across the semantics classes forms the basis of the semantics classification system.

Definition 2 CRT. *A composite region template* $\mathbf{T}$ *is an ordering of* M *symbols,* $\mathbf{T} = t_0 t_1 t_2 \ldots t_{M-1}$.

The region strings are consolidated by detecting and counting the frequencies of the CRTs in the set of region strings. For example, the test for $\mathbf{T} = t_0 t_1 t_2$ in region string $\mathbf{S}$ is given by $I(\mathbf{T}, \mathbf{S})$, where

$$I(\mathbf{T}, \mathbf{S}) = \begin{cases} 1 & \text{if } s_l = t_0 \text{ and } s_m = t_1 \\ & \text{and } s_n = t_2 \text{ and } l \leq m \leq n \\ 0 & \text{otherwise.} \end{cases}$$

The frequency of each CRT, $\mathbf{T}_i$, in a set of region strings $\{\mathbf{S}_j\}$ is then given by $P(\mathbf{T}_i)$, where

$$P(\mathbf{T}_i) = \sum_j I(\mathbf{T}_i, \mathbf{S}_j).$$

The frequency of each CRT, $\mathbf{T}_i$, in the set of region strings $\{\mathbf{S}_j\}_k$ from semantic class C_k is given by $P(\mathbf{T}_i|C_k)$, where

$$P(\mathbf{T}_i|C_k) = \sum_{\forall_j \mathbf{S}_j \in C_k} I(\mathbf{T}_i, \mathbf{S}_j).$$

4.2.3 CRT library

The CRTs derived from the training images construct the CRT library, which is defined as follows:

Definition 3 CRT library. *A composite region template library is given by a set of* $(K+2)$*-tuples:*

$$\{\mathbf{T}_i, P(\mathbf{T}_i), P(\mathbf{T}_i|C_0), P(\mathbf{T}_i|C_1), \ldots, P(\mathbf{T}_i|C_{K-1})\},$$

where K *is the number of semantic classes.*

4.3 Decoding image semantics

Once the CRT library is built from training images, it is used to semantically classify the unknown images. The semantics of an unknown image are decoded from its set of region strings using the CRT library as follows:

1. First, the region strings for the unknown image are extracted and consolidated into a set of CRTs.
2. For each CRT, $\mathbf{T}'_i$, from the unknown image, $P(C_k|\mathbf{T}'_i)$ is computed from the entries in the CRT library from:
$$P(C_k|\mathbf{T}'_i) = \frac{P(\mathbf{T}'_i|C_k)}{P(\mathbf{T}'_i)} P(C_k).$$
3. The classification of the unknown image is then given by: assign image to class l when
$$\forall_{l \neq k}, \quad \sum_i P(C_l|\mathbf{T}'_i) > \sum_i P(C_k|\mathbf{T}'_i). \tag{1}$$
That is, class C_l best explains the CRTs represented in the region strings of the unknown image.

4.4 Semantics classification evaluation

We evaluate the CRT-based semantics decoding method by measuring its performance in classifying unknown images from the eight semantic classes. Example images are illustrated in Figure 6. In the experiments, images from eight semantic classes were classified using the CRT method.

In total, 261 images were identified as belonging to the eight semantic classes. These 261 images were divided into non-overlapping training and test sets according to Table 3. The system used the 71 training images to generate the CRT library. The remaining 190 test images were used to evaluate the semantics classification performance of the system. The classification results are given in Table 3.

Given the eight semantics classes, the semantics decoding system using CRTs provides a classification rate of 0.784. The majority of classification errors resulted from a confusion between the buildings and nature classes. This is not surprising since both classes, as illustrated in Figure 6, often depict similar scenes, such as blue skies, above brown objects, above green grass.

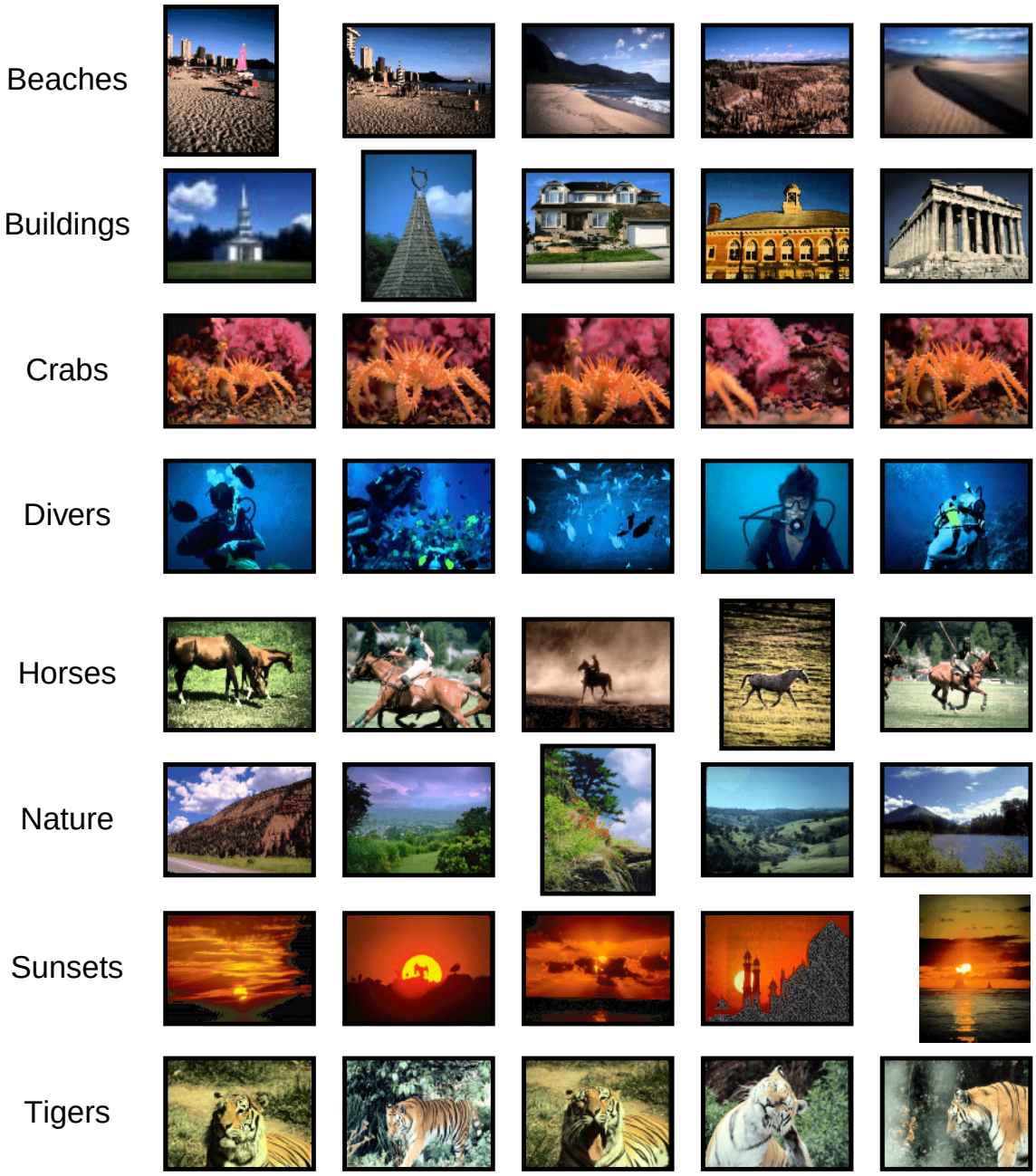


Figure 6: Example images from the eight semantics classes used to evaluate the CRT semantics decoding system: beaches, buildings, crabs, divers, horses, nature, sunsets, and tigers.

	overall	beaches	buildings	crabs	divers	horses	nature	sunsets	tigers
# total	261	14	56	9	33	26	46	46	31
# train	71	7	10	4	10	10	10	10	10
# test	190	7	46	5	23	16	36	36	21
# correct	149	6	30	5	23	14	20	31	21
% correct	78.4	85.7	65.2	100	100	87.5	55.6	86.1	100

Table 3: Image semantics classification experiment results using 71 training images and 190 test images from eight semantics classes.

5 Summary and Future Work

We presented a new system for classifying images using features and related text. The multi-stage image classification assigns the images to type, composition and semantics classes. Image type and composition are determined by mapping image features into a decision space that is adaptively partitioned using training images. Image semantics are determined by a novel system which matches the arrangements of regions in the images to composite region templates (CRTs). We developed a process by which this CRT library is constructed automatically from the images that are textually annotated.

We are applying the multi-stage image classification system to the classification of images on the World-Wide Web in order to better index and catalog this visual information. In particular, we are investigating the performance of the image semantics decoding system using several thousand semantics classes. Finally, we are exploring the utility of the image classification system for customizing the delivery of Web documents.

References

[ASF97] V. Athitsos, M. J. Swain, and C. Frankel. Distinguishing photographs and graphics on the World Wide Web. In *Proceedings, IEEE Workshop on Content-based Access of Image and Video Libraries*, June 1997.

[BFG+96] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu. Virage image search engine: an open framework for image management. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, volume 2670, pages 76 – 87. IS&T/SPIE, January 1996.

[FMF+96] D. A. Forsyth, J. Malik, M. M. Fleck, T. Leung, C. Bregler, C. Carson, and H. Greenspan. Finding pictures of objects in large collections of images. In *Proceedings, International Workshop on Object Recognition*. IS&T/SPIE, April 1996.

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23 – 32, September 1995.

[RF97] N. C. Rowe and B. Frew. Automatic caption localization for photographs on World Wide Web pages. Technical Report Code CS/Rp, Dept. of Computer Science, Naval Postgraduate School, 1997.

[SC96] J. R. Smith and S.-F. Chang. VisualSEEk: a fully automated content-based image query system. In *Proc. ACM Intern. Conf. Multimedia*, pages 87 – 98, Boston, MA, November 1996. ACM.

[SC97] J. R. Smith and S.-F. Chang. Visually searching the Web for content. *IEEE Multimedia*, 4(3):12 – 20, July–September 1997.

[Smi97] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, New York, NY, 1997.

The Theory and Practice of Similarity Searches in High Dimensional Data Spaces* Extended Abstract

Roger Weber
ETHZ
Zurich, Switzerland
weber@inf.ethz.ch

Pavel Zezula†
CNUCE-CNR
Pisa, Italy
zezula@iei.pi.cnr.it

August 27, 1997

1 Introduction

Similarity search in multimedia databases is typically performed on abstractions of multimedia objects, also called the features, rather than on the objects themselves. Though the feature extraction process is application specific, the resulting features are most often considered as points in high-dimensional vector spaces (e.g. the color indexing method of Stricker and Orengo [SO95]). Similarity (or dissimilarity) is then determined in terms of the distance between two feature vectors.

In order to manage similarity search retrieval in large object bases, several storage structures have been designed. However, most of the practical applications reported have observed the *dimensional curse*, i.e. the rapid performance deterioration with the increasing space dimensionality.

In this article, we elaborate on the performance issue of the similarity searches in high dimensional data spaces. Unless explicitly stated, the following assumptions are respected: (1) objects are described by feature vectors in a d-dimensional vector space, (2) the similarity is measured as the Euclidean distance, (3) there is no correlation between data

*This research has been funded by the EC ESPRIT Long Term Research program, project no. 9141, HERMES (Foundations of High Performance Multimedia Information Management Systems). The work of Pavel Zezula has also been supported by Grants GACR No. 102/96/0986, Object-oriented data model, and KONTAKT No. PM96 S028, Parallel text bases. The work of Roger Weber has been funded by the Swiss *Bundesamt für Bildung und Wissenschaft* (BBW, grant no. 93.0135).

†On leave from the CVIS, Technical University, Údolni 19, Brno, Czech Republic, E-mail: zezula@cis.vutbr.cz

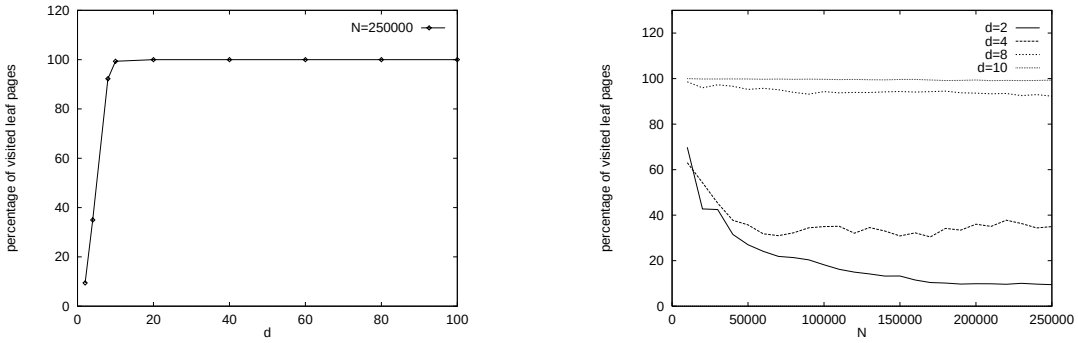


Figure 1: Nearest Neighbor search in R*-Trees

on different levels, and (4) the feature vectors are uniformly distributed in the data space $D = [0, 1]^d$.

Given these assumptions, we start with some experiments with R-Tree-like methods [Gut84, BKSS90, BAK96] to confirm the fact, that they completely fail, if the dimensionality goes beyond a small number, say 16.

To better understand why these methods fail, we investigate the nearest neighbor search from a theoretical point of view. Then, recent proposals, based on different partitioning principles, are considered and their performance characteristics investigated with respect to the feature vector dimensionality.

2 The dimensional curse for the R-Tree structures

This section presents some experiments with the R*-Tree [BKSS90], followed by theoretical studies on the nearest neighbor search performance, in general.

2.1 Experiments with R*-Tree

The R*-Tree [BKSS90] is an improved access method of the R-Tree [Gut84]. Both methods partition the data space recursively and store information about the partitions in the nodes. The partitions are described by the minimal bounding box, which covers all objects within the partition. The R-Tree typically consists of a small number of levels (< 10) and a high fill grade of the leaf nodes ($> 70\%$). A provable optimal nearest neighbor search algorithm has been defined by Hjaltson and Samet [HS95] and proved by Berchtold et al. [BBAK97]. It visits the partitions according to their minimal distance to the query point. As soon as a vector has been found, that lies nearer to the query point than all remaining partitions, the nearest neighbor has been found. Generally, the algorithm stops after having found k points—the k nearest vectors to the query. To measure the quality of an access method, one typically counts the number of visited pages. Given the tree structure of the R*-Tree, we assume, that all but the leaf nodes are cached in memory.

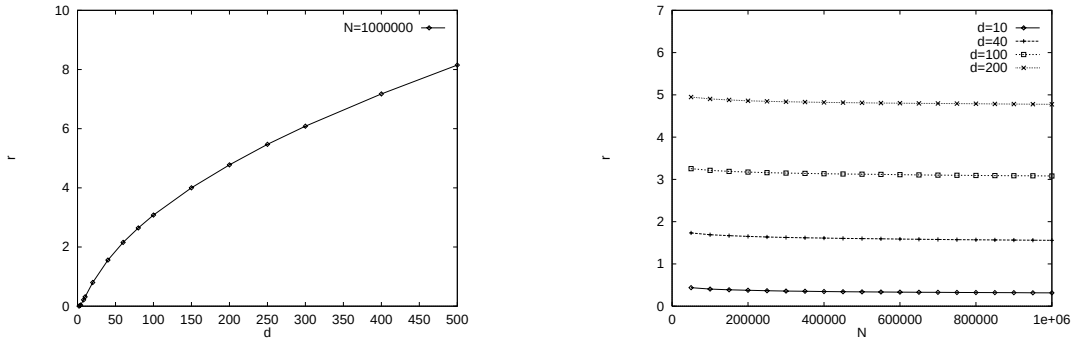


Figure 2: Expected nearest neighbor distance

Therefore, we only measure the number of visited leaf nodes.

Figure 1 shows the number of visited leaf nodes of an R*-Tree for an uniformly distributed data set within the vector space $D = [0, 1]^d$, whereas d is the dimension of the space and N is the number of points in the database. The page size for all trees was 8K. One quickly observes, that nearest neighbor search in R*-Tree is hopeless as soon as the dimensionality goes above 10, because all leaf nodes must be visited. Consequently, a simple scan through the vectors would perform better than a search with the R*-Tree.

2.2 Theoretical Studies of the Nearest Neighbor Problem

Similar to [BBAK97], we first computed the expected nearest neighbor distance in high-dimensional vector spaces (see Figure 2). As expected, the distance grows as dimensionality gets larger and shrinks if more points are used.

Then, we computed the number of leaf pages intersecting the query sphere, that is the sphere around the query point with the radius equal to the nearest neighbor distance (see Figure 2). For this purpose, we have to determine the Minkowski sum of the query sphere and the bounding box of the leaf pages. Instead of evaluating the closed formula given in [BBAK97], we focused on the size of the bounding boxes at the leaf nodes. We found out, that the concept of Minkowski sum transforms the leaf pages in enlarged objects, which cover the entire data space. In other words, each of the leaf nodes intersects the query sphere and therefore has to be visited during a nearest neighbor search [1]. To verify this statement, we examined the leaf nodes of R*-Trees and computed the maximal distance of their bounding boxes to any point in the data space. The results together with the expected nearest neighbor distance are shown in Figure 3. The fact, that the maximum distance to any point in the data space is smaller than the expected nearest neighbor distance proves, that the enlarged leaf page incorporates the entire data space. Consequently, all leaf nodes of the R*-Tree must be visited during a nearest neighbor search and the search degrades to a linear problem. Further experiments have shown, that this holds true for every tree-like structure, which uses hyper cubes as bounding

[1] The prove for this and the following statements is subject of a future paper

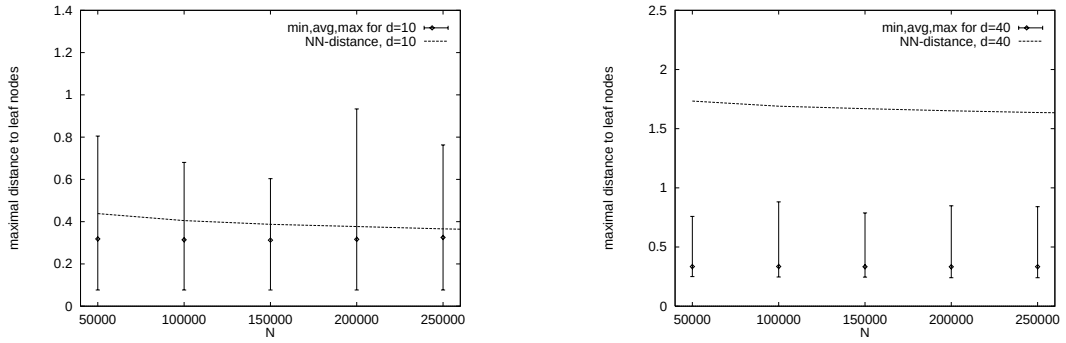


Figure 3: Nearest neighbor distance compared to the maximal distance to leaf nodes; on the left for $d = 10$ and on the right for $d = 40$

boxes.

In literature, it is commonly accepted that the X-Tree [BKSS90] is the most efficient tree structure for nearest neighbor search in high-dimensional vector spaces. Figure 1 and several experiments [BKSS90, BBAK97, BBB+97] show, that the X-tree is better than the R*-Tree, but can not get rid of the dimensional curse too. As long as the dimensionality is low (less than 16) the X-tree can efficiently prune the search space for nearest neighbor queries, but for vectors of higher dimensionality, the complexity of nearest neighbor search becomes $O(n)$, because all of the leaves must be visited.

3 New Approaches

As we have just demonstrated, the nearest neighbor problem is, for uniformly distributed high-dimensional vector sets, linear. Furthermore, experiments have shown that a simple scan through the database outperforms more sophisticated tree-like structures. However, it is possible to reduce the cost of the sequential scan by using small approximations of the vectors, which are stored in a separate, much smaller, file. These approximations may be used to underestimate the distance of the object to the query and to filter out candidates. This method works like the signature method and is called *VA-File*.

The second approach, the so called M-tree, enables nearest neighbor search for metric spaces, which includes the previously discussed vector spaces. The principle difference with this method is that the search space is partitioned only according to distances between objects, i.e. no coordinates of the space are used. From the performance point of view, M-tree aims at reducing not only the I/O, but also the CPU costs.

3.1 The VA-File

Similarly to signature methods, the vector approximation file (*VA-File*) proposed here does not partition data as the R-Tree like methods do. In particular, the data space is

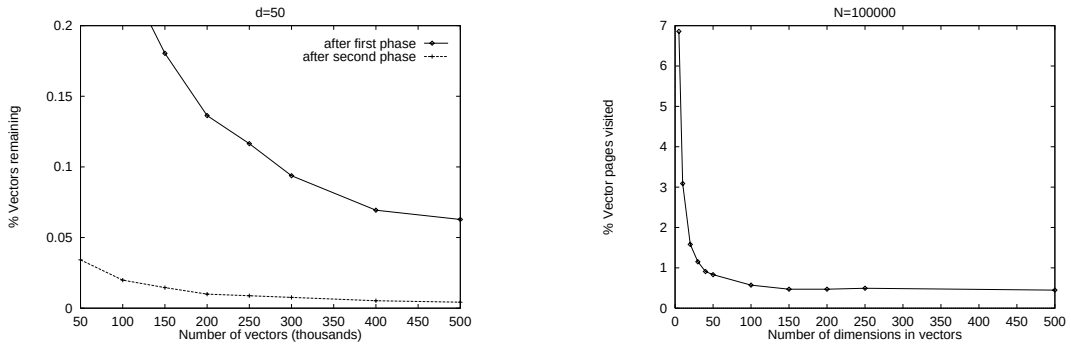


Figure 4: Selectivity as a function of the databases size (left); and page selectivity as a function of d

first partitioned into regions, and then these regions are used to generate bit-encoded approximations for each vector. Contrary to the grid-file [NHS84], no directory over these cells is built, because the number of cells grows exponentially with dimensionality. Rather, all approximations are stored in a sequential file. Queries, vectors in the same data space, are not approximated. To perform nearest neighbor search, the VA-File operates in two phases. In the first phase, it applies a filter. Based on the approximation, it computes for each approximation the minimal and maximal bound on the distance between the query point and the region represented by the approximation. Given that one is looking for the k nearest vectors to the query point, one can select an approximation as a candidate for the second phase if its minimal bound is smaller than the maximal bound of the k-th nearest approximation. In the second phase, the candidates are visited in increasing order of their minimal bound to determine the final answer set. This phase ends when a minimal bound is found which exceeds or equals the k-th best distance in the answer set. Our practical experiments have shown, that between 95% and 99% of the vectors are eliminated during the first filtering step, and 99.9% over all (see Figure 4, left side). Thus, only a small number of vectors must be accessed eventually, and the total number of I/O operations is smaller than with sequential scanning all vectors.

Figure 4 (left) shows the effectiveness of the two filtering steps with vectors of 50 dimensions. The right side of Figure 4 proves, that the VA-File does not suffer from the dimensional curse (the number of vectors is allways 100000). It even gets better with growing dimensionality. Furthermore, in wall-clock experiments, the VA-File was up to three times faster than a simple scan. The larger the database was, the better the method performed.

3.2 The M-tree

In [PP97], a paged metric tree, called M-tree, has been designed. In order to organize and partition the search space, this approach only considers relative distances, rather than absolute positions, of objects in a multi-dimensional space. The M-tree is a balanced tree, able to deal with dynamic data files, and as such it does not require periodical reor-

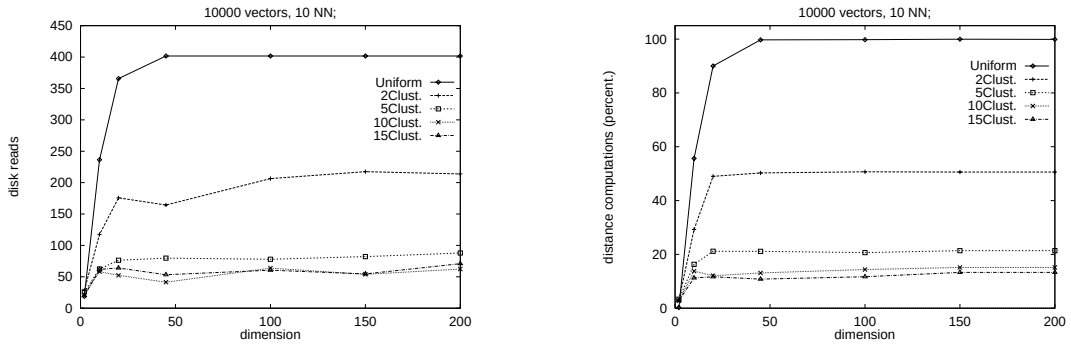


Figure 5: Selectivity as a function of the databases size (left); and page selectivity as a function of d

ganizations. M-tree can also index objects using features compared by distance functions which either do not fit into a vector space or do not use an L_p metric, thus considerably extends the cases for which efficient query processing is possible. What is only required for this approach to work is that the function used to measure the distance, or better to say the dissimilarity, between objects is a *metric*, so that the *triangle inequality* property applies and can be used to prune the search space. Naturally, the M-tree can support both the range and the k-nearest neighbor queries.

As expected, M-tree does suffer from the dimensional curse provided the data is independent and uniformly distributed in n-dimensional space. With such files, the 10 nearest neighbor (NN) queries on 10-dimensional vectors already require more than 50% of both the page reads as well as the distance computations – vectors of 20 dimensions push this percentage up to 90%, and the search is becoming practically linear for $n = 25$. The results are presented in Figure 5.

However, interesting behaviour can be observed for "clustered" feature files – real life files are rarely uniform and independent, thus clusters of objects can be recognized (see Figure 5). With only just five clusters, the number of necessary distance computations has been upper bounded by about 20% of the vectors in the file, and this was true for any file with dimensionalities between 20 and 200 – 200 dimensions was the maximum dimensionality we have tried. Similar behaviour has been observed for the necessary number of page reads.

4 Conclusions

The performance problem of similarity searches in high dimensional data spaces has been elaborated. It has been demonstrated, both by experiments and the theoretical analysis, that the tree oriented search structures used for similarity retrieval significantly suffer from the dimensionality curse, provided the vectors are independent and uniformly distribute. Since, in such situation, the performance complexity seems to be linear, the sequential search on the vector file, or, better, on its approximation, such as the VA-file, seems to

be the best solution.

However, as soon as the feature vectors start to form clusters, the performance of tree-based structures can significantly improve. At least such behaviour has been observed for the M-tree.

References

[BAK96] Stefan Berchtold, Daniel A.Keim, and Hans-Peter Kriegel. The X-tree: An index structure for high-dimensional data. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 28–39, 1996.

[BBAK97] Stefan Berchtold, Christian Böhm, Daniel A.Keim, and Hans-Peter Kriegel. A cost model for nearest neighbour search. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 78–86, Tucson, Arizon USA, 1997.

[BBB+97] Stefan Berchtold, Christian Böhm, Bernhard Braunmüller, Daniel A.Keim, and Hans-Peter Kriegel. Fast parallel similarity search in multimedia databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12, Tucson, Arizon USA, 1997.

[BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In Hector Garcia-Molina and H. V. Jagadish, editors, *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, NJ, 23–25 May 1990. *SIGMOD Record* 19(2), June 1990.

[Gut84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In B. Yormack, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, Boston, MA, June 1984. acm.

[HS95] G. R. Hjaltason and H. Samet. Ranking in spatial databases. *Lecture Notes in Computer Science*, 951:83–??, 1995.

[NHS84] J. Nievergelt, H. Hinterberger, and K. C. Sevcik. The grid file: An adaptable symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38–71, March 1984.

[PP97] M.Patella P.Ciaccia and P.Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, Athens, Greece, 1997.

[SO95] Markus Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases, SPIE*, San Jose, CA, 1995.

The DELOS Working Group is a part of the ESPRIT Long Term Research Programme (LTR No. 21057) and is managed by ERCIM.

The DELOS Partners are:

ERCIM members:

CLRC, CWI, CNR, FORTH, GMD, INRIA, INESC, SICS, ETH-SGFI, SINTEF Telecom and Informatics, MTA SZTAKI, VTT

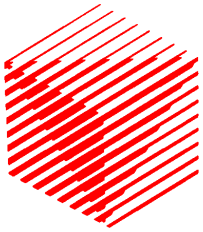
Non-ERCIM members:

University of Michigan, USA
Elsevier Sciences, The Netherlands

For additional information, please contact
Costantino Thanos
Istituto di Elaborazione della Informazione, Consiglio Nazionale delle Ricerche
Via Santa Maria 46
I-56126 Pisa
Tel: +39 50 593492, Fax: +39 50 554342, E-mail: thanos@iei.pi.cnr.i

DELOS web site: **http://www.area.pi.cnr.it/ErcimDL/delos.html**

ISBN 2-912335-03-5



The European Research Consortium for Informatics and Mathematics (ERCIM) is an organisation dedicated to the advancement of European research and development, in the areas of information technology and applied mathematics. Through the definition of common scientific goals and strategies, its national member institutions aim to foster collaborative work within the European research community and to increase co-operation with European industry. To further these objectives, ERCIM organises joint technical Workshops and Advanced Courses, sponsors a Fellowship Programme for talented young researchers, undertakes joint strategic projects, and publishes workshop, research and strategic reports as well as a newsletter.

ERCIM presently consists of fourteen research organisations from as many countries:

Central Laboratory of the Research Councils

Rutherford Appleton Laboratory
Chilton, Didcot
GB-Oxon OX11 0QX

Tel: +44 123582 1900
Fax: +44 1235 44 5385
http://www.cis.rl.ac.uk/

Centrum voor Wiskunde en Informatica

Kruislaan 413
NL-1098 SJ
Amsterdam

Tel: +31 205929333
Fax: +31 20 592 4199
http://www.cwi.nl/

Consiglio Nazionale delle Ricerche

IEI-CNR
Via S. Maria, 46
I-56126 Pisa

Tel: +39 50 593 433
Fax: +39 50 554 342
http://bibarea.area.pi.cnr.it
/ERCIM/welcome.html

Czech Research Consortium for Informatics and Mathematics

FI MU
Botanicka 68a
602 00 Brno

Tel: +420 2 6884669
Fax: +420 2 6884903
http://www.utia.cas.cz/
CRCIM/home.html

Danish Consortium for Information Technology

CIT
Forskerparken
Gustav Wieds Vej 10
8000 Århus C

Tel: +45 8942 2440
Fax: +45 8942 2443

Foundation of Research and Technology – Hellas

Institute of Computer Science
P.O. Box 1385
GR-71110 Heraklion, Crete

Tel: +30 81 39 16 00
Fax: +30 81 39 16 01
http://www.ics.forth.gr/

GMD – Forschungszentrum Informationstechnik GmbH

Schloß Birlinghoven
D-53754 Sankt Augustin

Tel: +49 2241 14 0
Fax: +49 2241 14 2889
http://www.gmd.de/

Institut National de Recherche en Informatique et en Automatique

B.P. 105
F-78153 Le Chesnay

Tel: +33 1 39 63 5511
Fax: +33 1 39 63 5330
http://www.inria.fr/

Instituto de Engenharia de Sistemas e Computadores

Rua Alves Redol 9
Apartado 13069
P-1000 Lisboa

Tel: +351 1 310 00 00
Fax: +351 1 52 58 43
http://www.inesc.pt/

Swedish Institute of Computer Science

Box 1263
S-164 28 Kista

Tel: +46 8 752 1500
Fax: +46 8 751 7230
http://www.sics.se/

Schweizerische Gesellschaft zur Förderung der Informatik und ihrer Anwendungen

Math. Dept.,
ETH-Zentrum
CH-8092 Zürich

Tel: +41 1 632 22 25
Fax: +41 1 632 10 85
http://www-dbs.inf.
ethz.ch/sgfi/

Stiftelsen for Industriell og Teknisk Forskning ved Norges Tekniske Høgskole

SINTEF Telecom &
Informatics
N-7034 Trondheim

Tel :+47 73 59 30 00
Fax :+47 73 59 43 02
http://www.informatics.
sintef.no/

Magyar Tudományos Akadémia – Számítástechnikai és Automatizálási Kutató Intézete

P.O. Box 63
H-1518 Budapest

Tel: + 361 166 5644
Fax: + 361 166 7503
http://www.sztaki.hu/

Technical Research Centre of Finland

VTT Information Technology
P.O. Box 1200
FIN-02044 VTT

Tel:+358 9 456 6041
Fax :+358 9 456 6027
http://www.vtt.fi/