

Semantic Content Based Image Retrieval Using Object-Process Diagrams

Dov Dori¹ and Hagit Hel-Or²

¹ Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
Haifa 32000, Israel

² Department of Mathematics and Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel

Abstract. The increase in accessibility to on-line visual data has promoted the interest in browsing and retrieval of images from Image Databases. Current approaches assume either a text based key-word oriented approach or a visual feature based approach. The keyword approach usually provides neither layout information nor object relevance and significance in the scene. The visual feature based approach relies on low level features such as color, texture and orientation as image descriptors. These are non-intuitive and unnatural for human observers. This paper presents a new approach to image retrieval in which image content, based on the "visual scene", is the basis for both retrieval and user interface. We propose to model image content using Object-Process Diagrams. Our hierarchical approach incorporates both the low-level image features and textual key sentences as descriptors of the image. These descriptions involve the objects in the scene and their inter- and intra-relationships. This allows for abstract, high-level representation of the layout of the scene, as well as a distinction between the dominant core of the scene and its background. Querying is performed by representing the sought image with an Object-Process Diagram and finding the images in the database whose Object-Process Diagrams best match the query.

1 Introduction

With the increase of quality and availability of on-line communications, enormous quantities of visual data have become accessible over the network. Additionally, databases of visual data are now being integrated into many systems and applications. These advances call for facilities for on-line browsing and searching of visual information.

The large volume of available visual data and the distribution of this data in an increasing number of sites makes sequential serial search impractical, and in fact, impossible. This stimulates the study and development of automatic and interactive computer-assisted tools for retrieval of visual data.

Most current approaches to image retrieval employ either a text based key-word oriented approach or a visual feature based approach.

In the **Keyword oriented data search**, keywords are used to describe the visual data textually. These keywords are usually determined manually by a viewer and are linked to the data (usually by insertion into the image header, or visual data attribute list). Image retrieval is then performed using text search over the keywords associated with the visual data. [18].

In the **Visual feature based approach**, low level visual features are used as search keys into the visual database. These features include color, texture and possibly orientation cues. The interface supplied to the user in the retrieval tool usually takes the form of the user marking regions in the image with the desired low level features of color and texture.

In the keyword approach, the retrieval based on keyword search is restrictive and is therefore expected to retrieve too wide a range of selections, since keywords, though informative, usually do not provide layout information and do not represent relevance and importance (dominance) in the scene. For example, we can expect both Figure 1a and Figure 1b to be associated with the keyword "apple" although it is obvious that in Figure 1a this keyword is much more significant to the image content than it is in Figure 1b. In some sense, keywords describe portions of the image content, but frequently fail to capture the visual scene.

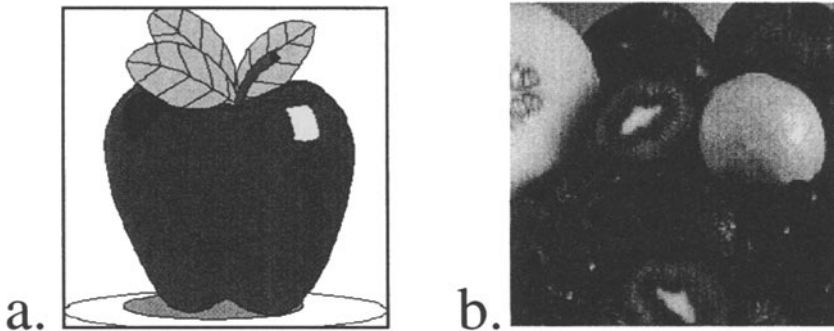


Fig. 1. Keywords do not capture significance in the image content. The keyword "apple" would be associated with both images a and b, however it has more significance in a than in b.

In the visual feature approach, the interfaces supplied to the user are non-intuitive and unnatural. The user searching for visual data usually has some idea of the image content and the image layout of the desired image, where the image content is usually described in terms of objects and global features, rather than low level features, such as color and texture. Additionally, abstract characteristics of the image and overall effects (such as mood, lighting, coloring etc.) often serve as descriptors of desired images and should be available as keys

for search in the visual database. Low level features are inefficient as descriptors of visual objects as they do not capture the abstract characteristics of the image, which are typically the main source of interest to the viewer.

We propose an intuitive and natural approach to image retrieval, in which image content, based on the visual scene, is the basis for retrieval and for the user interface. This approach is superior to both the non-intuitive approach of low level features as search keys and the keyword search, which does not capture the structure and dynamics of the image scene. The approach we propose has hierarchical characteristics. It also incorporates both the low-level image features and textual keywords as descriptors of the image.

We propose to model image content using Object-Process Diagrams (OPDs). These graph-like descriptions involve the objects in the scene and their inter- and intra-relationships. This allows for abstract, high-level representation of the layout of the scene, as well as a distinction between the dominant core of the scene and its background. This representation inherently includes textual keywords as object names and low level features as object attributes.

2 Previous Work in Image Retrieval

The large quantities of visual data that are now easily accessible make exhaustive search of images impractical. Automated and computer-assisted search of visual data is becoming a necessity. The interest of the research community in this topic is therefore constantly increasing in this domain.

A typical system assumes a large image database, which may be from a specific source, with constrained content, or a general, unrestricted database. Images are retrieved from the database using keys (which might be textual, visual or symbolic - see below). Typically, the user submits a query based on the available keys, preferably through a user-friendly interface tool. The output of the retrieval system is usually a ranked set of images from the database, ordered by the likelihood of answering the query correctly. The likelihood is determined by a similarity measure which is used to compare the query with the images in the database. Many image retrieval systems have been suggested. They vary in the type of keys, the query interface-tool and the similarity measure.

The most basic retrieval system, based on extensions of textual database search, is a system in which the keys are keywords annotating the images and usually describing certain aspects of the image content [18]. Queries are performed using standard query languages such as SQL. Keyword indexing is very restrictive, as it generally does not capture the scene layout nor the relationships between objects. It also does not distinguish between central elements in the scene and minor parts (such as foreground and background). Additionally, some visual elements are hard to describe using only text [26]. Recent work extends the text-based image retrieval to be hierarchical and more flexible [27].

More common approaches to image retrieval are the content-based image retrieval systems, in which the keys are low-level visual features such as color, shape, orientation, texture etc. Typically, coding of the visual feature is defined,

which is invariant to location in the image. The user generates a query, which is also a representation of a visual feature. A similarity measure for comparing query and images is developed specifically to deal with the coded visual feature. Retrieval of images, based on color, usually represents this feature using color histograms [35, 26, 15] and the similarity measure is based on comparing histograms. This type of representation tends to produce false positives. Color histograms give statistics of the image pixels but do not provide spatial, relational or content information in terms of objects in the scene. To improve performance, locational information was incorporated into the color-based image retrieval system by allowing multiple color histograms, representing different locations in the image [26, 15] and using color histograms containing spatially dependent classes [28]. Combining spatial and color information using a Markov model was also suggested [22]. Improvements of the color-based image retrieval approach were recently suggested: multiresolution and different colorspace were tested [39], while compression and efficient representation of local color histograms was suggested in [41].

The user interface for producing the visual query usually takes the form of painting regions of a blank image with the sought color (average color of a region) [26], by sketching in color the general outline of the sought image [3, 16], or by providing an example image [41, 29]. Describing an image by regions of color is unintuitive and unnatural. Sketching a query or providing an example implies strict positional information, which is not necessarily desired. Additionally, this representation allows neither variability of the objects in the scene nor flexibility in the relationships among them. Implicitly allowing flexibility in location and shape may help [2], as we describe below.

Shapes as low-level visual features have been used as keys for image retrieval. Classical studies on object and pattern matching can be reformulated for use in shape-based image retrieval. For example, Geometric Hashing was extended for image retrieval [37], a two stage refinement procedure was used in [38], and edge based matching was used in [16]. Shape representation as keys for image retrieval usually requires a segmentation process or an edge detection process, which introduces variance, if not errors, in the shape representation. Shape descriptors based on radial descriptors that do not need segmentation or edge detection was suggested in [1].

Texture serves as a key for image retrieval in several studies [19, 21, 29]. Scale and orientation selective filters determine texture parameters in [23]. An efficient parallel approach to texture classification for image retrieval was suggested in [42].

The various low-level visual features, used as keys, have been combined in image retrieval systems, either as combined features [20], or as separate modules [26, 29, 14].

Several other approaches to image retrieval have been suggested, including the use of transform coefficients as keys [40, 34, 32], using eigenfeatures [29, 36] and image retrieval from a compressed database [33, 40]. Locational information between objects in a scene has been incorporated into an image representation

using 2D strings [31] and 2D Markov Models [22]. A review on image retrieval can be found in [24, 30].

3 Object-Process Methodology and Diagrams

Any system has two major aspect: structure and behavior. Structure pertains to relationships among things (objects or processes) in the system that hold in the long run, while behavior has to do with the dynamics of the system, i.e., the way its state changes over time. The Object-Process Methodology (OPM), first introduced in [6, 10], is an integrated approach to the development of systems that unifies structure and behavior throughout the analysis, design and implementation of the system within one frame of reference using a single diagramming tool - the Object-Process Diagram (OPD) [5]. The basic premise of OPM is that objects and processes are two types of equally important classes of things, that together faithfully describe both the structure and the behavior of systems in virtually any domain. The major difference between OPM and current "classical" Object-Oriented (OO) development methods is that while OO methods employ a host of models, each with its diagramming symbols and conventions, to describe the various system aspects, OPM uses a single object-process model, with the OPD set as its single graphic modeling tool. This eliminates the model multiplicity problem, which requires mental integration of the various models into a coherent understanding of the system under consideration. OPM also features a rich set of scaling tools: blow-up, unfolding and explosion, which provide for flexible, yet consistent complexity management through selectively controlling the visibility of system details.

3.1 OPM Applications

OPM is generic in nature, as it is founded on principles of systems theory. It has been employed in a variety of domains, including engineering drawing understanding [4, 8], 3-D object reconstruction [13], analyzing R&D of high-tech firms [25], image understanding [9] and computer integrated manufacturing [7]. OPM encompasses not just the analysis phase of systems development, but also the design and implementation phases [12, 11].

OPCAT (acronym for Object-Process CAse Tool) has been developed as the Computer Aided Software Engineering (CASE) tool to support the Object-Process Methodology. Since 1994 OPCAT has evolved from a modest program to a semi-commercial product with version control and configuration management.

4 Visual Object-Process Diagrams - VOPD

A Visual Object-Process Diagram (VOPD) is a specialized OPD designed to describe visual scenes that appear in images. One of the initial steps in our research is to specialize general purpose Object-Process Diagrams to Visual

Object-Process Diagrams, so that they become suitable for representing images. Basic to a generic VOPD is the distinction between foreground and background. This distinction is primarily a content-based, or semantic observation. In terms of the Object-Process Methodology, there is an aggregation relation between the image as a whole and its foreground and background parts. In the generic VOPD, shown in Figure 2, the aggregation is denoted by a black triangle. The structural relation between the two parts is that the foreground is in front of the background. The focus of interest in an image is usually the objects in the foreground, but frequently the background is also an important criterion for image retrieval. The image as a whole has a number of low-level attributes, including name, size, gray-level or color histogram, etc. These low-level features, along with keywords, are used in current image retrieval systems. The problem with low-level features is that they do not relate to the content of the objects represented in the image, and hence the retrieved images usually contain images that are far from what the user expected to find. By attaching a VOPD to the image, we establish reference to the semantic content of the image, rather than to its appearance. This approach is novel and unique in that it relates to the semantic content that the image shows, rather than to how it is shown. This by no means implies that low-level features are not addressed. On the contrary, by referring separately to objects in the foreground and in the background, we can attach different low level attributes to each one of them. Thus we can formulate a query in which an object in the background is an instance of class X (e.g., X=House) and the dominant color attribute of the object is white, while the foreground contains an object instance of class Person and its dominant color attribute is red. Thus, the interplay between the semantic image content – the high-level cognitive aspects of what the image expresses – is combined with low-level attributes of objects in the image to yield a query that is much more specific and accurate than what state-of-the-art image retrieval systems produce.

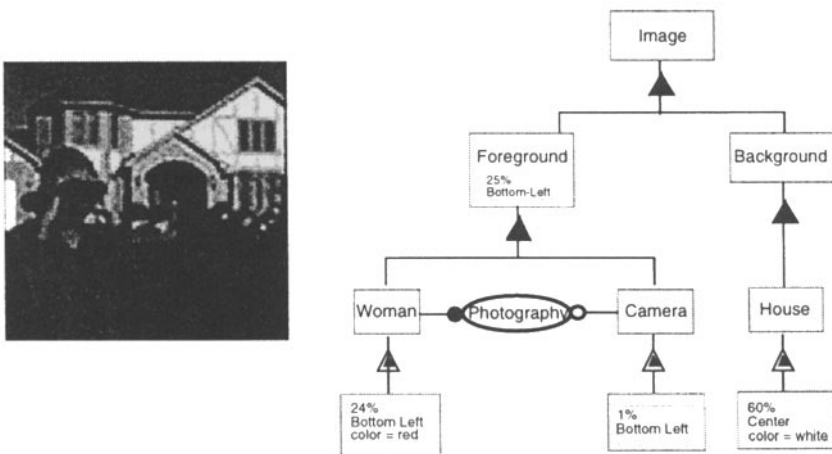


Fig. 2. An image (left) represented by a Visual Object-Process Diagram (right).

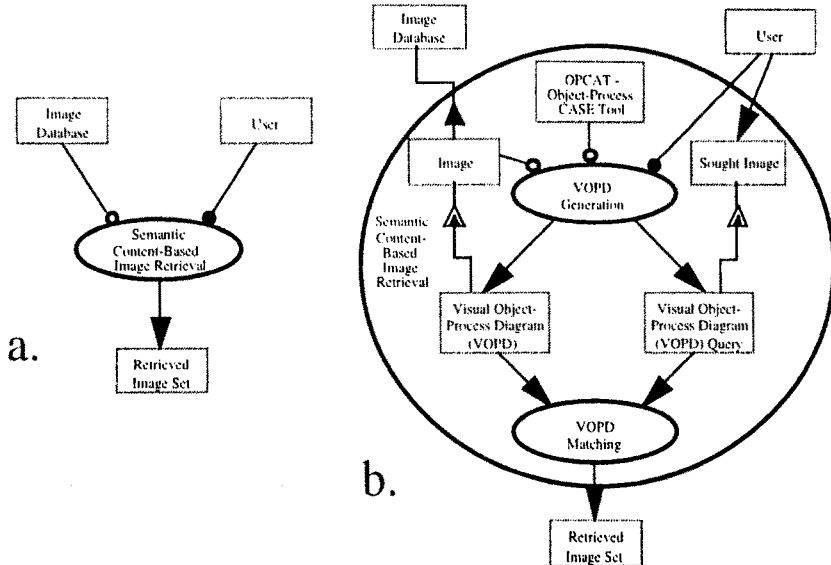


Fig. 3. A set of Object-Process Diagrams describing the Semantic Content Based Image Retrieval System. a) Top-level OPD b) The process Semantic Content Based Image Retrieval - blown up.

5 Semantic Content Based Retrieval - Overview

In our scheme, images of visual scenes are represented in terms of the objects in the scene, their relationships (in terms of positions and actions), local (object-based) attributes and global (scene-based) attributes. These are all expressed through the Visual Object-Process Diagram (VOPD).

Each image in the database is indexed by an associated VOPD. Retrieval from the database is based on a measure of similarity between sought and existing VOPDs. This measure considers distances between object-process diagrams while taking into account the fact that these represent visual data.

Figure 3a shows a top-level Object-Process Diagram describing the Semantic Content Based Image Retrieval System. In Figure 3b, the main process of our system - Content Based Image Retrieval- has been blown up. As can be seen, this process consists of two main low-level processes: *VOPD Generation* and *VOPD Matching*. The process of VOPD Generation is applied off-line to all images in the database associating a VOPD to each image. It is applied on-line during image retrieval to generate a query VOPD from the user input. The process of VOPD Matching is applied during image retrieval and involves comparisons between the query VOPD and the VOPDs associated with the images in the database. The output of this process is a (ranked) set of images whose VOPD match the query VOPD to a pre-defined degree.

6 Semantic Content Based Image Retrieval - Example

As an example for our approach, we created VOPDs for several images. Figure 4 and Figure 5 show the images (left) and their associated VOPDs (right). These images and their associated VOPDs will be denoted the Image Database. Several query VOPDs were created either abstractly or from other images. The images in the Image Database were manually ranked according to the expected distance between the associated VOPD and the query VOPD.

Obviously, retrieval based on keywords, such as “Apple”, will extract the three images of Figure 4a-c. However the image content and layout of these images differ significantly. Accordingly, the associated VOPDs are also different. Thus retrieval based on VOPD (which includes an “Apple” object) will perform a more precise retrieval.

Another example is shown in Figure 6a, where a query image and its associated VOPD are shown. Although all three images of Figure 5 contain the keyword “Lake”, Figure 5c would rank the highest with respect to the query. Indeed this image is visually similar to the query image in terms of image content and scene layout.

Figure 6b shows a query image whose local average color is exactly the same as that of the image in Figure 5c. Retrieval based on low level features such as local color, would rank Figure 5c as very similar to the query image. Based on Semantic content based retrieval, these two images are very different as can be seen from their associated VOPDs.

The examples presented here, serves to show the capabilities of our approach and to point out the advantages of our approach over previous studies in image retrieval.

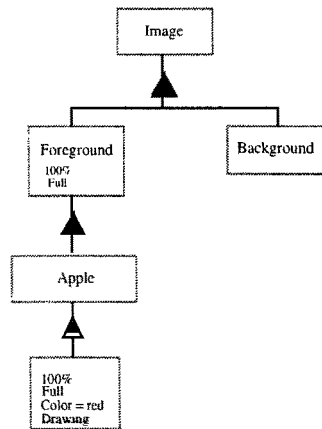
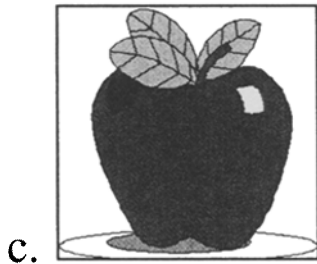
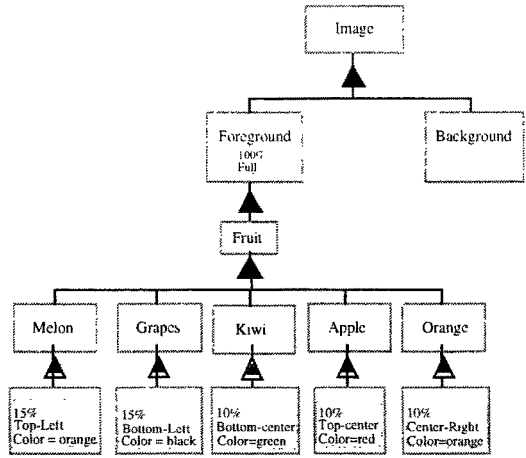
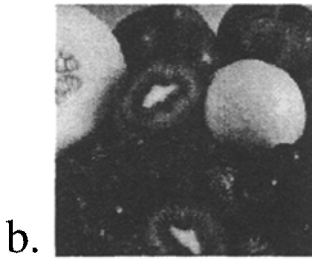
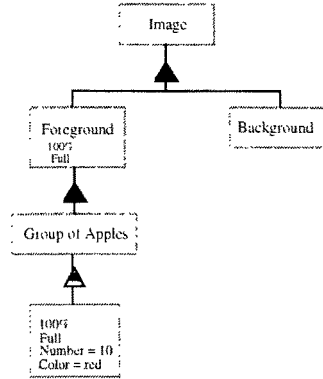
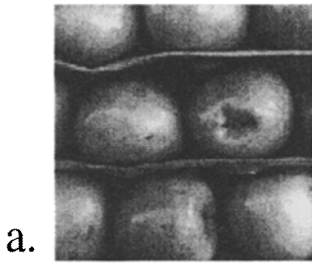
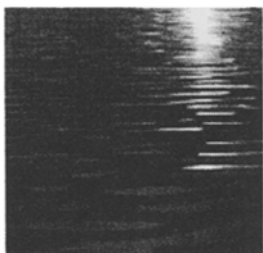


Fig. 4. Three images (left) and their associated Visual Object-Process Diagrams (right).

a.



b.



c.

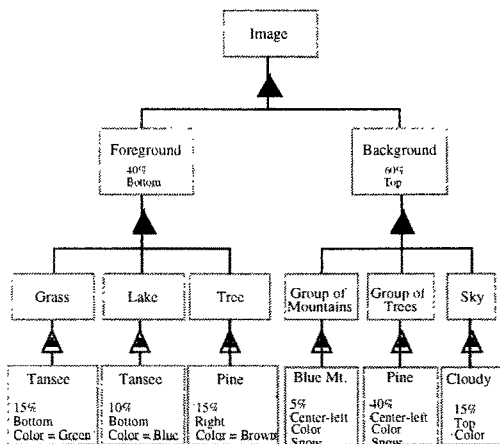
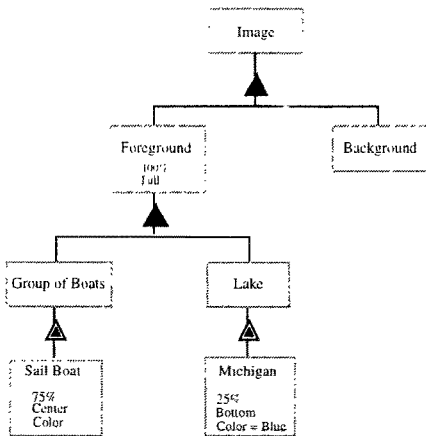
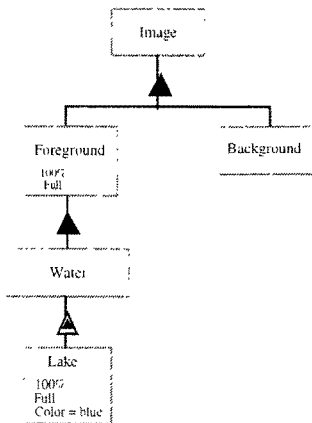


Fig. 5. Three images (left) and their associated Visual Object-Process Diagrams (right).

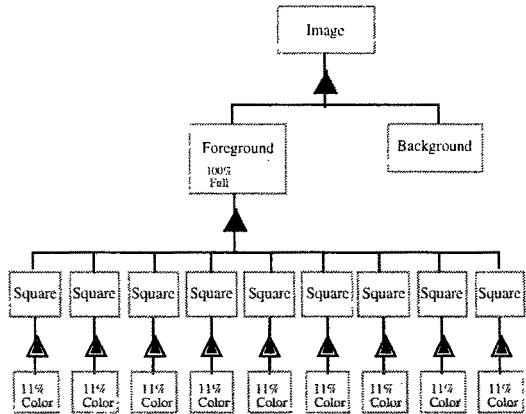
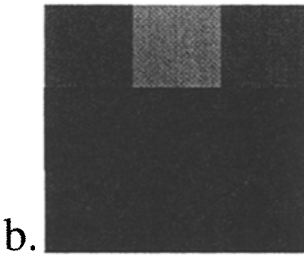
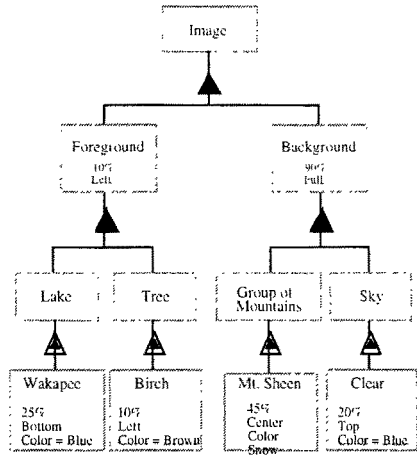
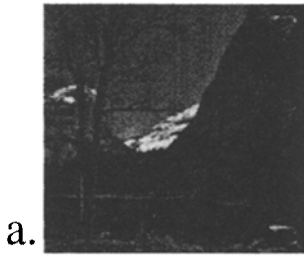


Fig. 6. Query images (left) and their associated Visual Object-Process Diagrams (right).

7 VOPD Generation

The VOPD Generation Process consists of the following lower-level processes:

1. Extraction of objects and visual components from an image to be included in the VOPD: This process includes extraction of objects and their inter-relations, understanding of dominance in the scene, determining foreground and background and possibly determining some overall attributes of the scene, such as mood, time of day, lighting conditions etc. Extraction of the image components may be manual or automatic. Several independent modules may be used to perform parts of this task. These include a segmentation module, object recognition modules, and illumination understanding modules. As additional modules become available, they can be incorporated into this part of the process.
2. Creation of VOPDs from the visual data: Once the objects have been extracted, the actual VOPD representing the visual data is created by a specialized version of OPCAT, shown as the tool for the VOPD Generation process in Figure 3b. Creation of the VOPD should follow the syntax and semantics of OPM.
3. Creating a query VOPD from the user's input using the same interface tool - OPCAT: OPCAT allows the user to define objects in the image, actions, relationships between objects, locational information (between objects and relative to the image frame), object attributes and general attributes of the image (such as mood, lighting etc). Given the user's input, an appropriate VOPD is created. Modules developed for the automatic generation of VOPDs from visual data, (above) can be exploited here as well.

8 VOPD Matching

Comparison between VOPDs is performed with an understanding that they represent visual data. Thus flexibility must be provided in the comparison to allow for variability in the visual scene in terms of object-based image content. The comparison must take into account the dominant core of the visual scene and weigh it appropriately. Background and less dominant scene features should have less of an affect on the comparison. The hierarchical nature of the VOPD should be taken into account and should assist the matching process. The comparison may have user defined parameters associated with it, to guide and emphasise certain aspects of the comparison. For example, illumination or layout may be more important than the actual objects and actions in the scene.

The final output of a comparison between VOPDs is a measure of similarity which can be used to rank order visual data in the database and respond to the user's query by displaying retrieved images in a decreasing similarity order.

The comparison among VOPDs comprises several levels of matching:

- **Unstructured Matching** - matching between two VOPDs is based on object names appearing in each representation. This type of matching follows

the keyword based approach to image retrieval. A list of keywords is composed for each VOPD by extracting the object names and possibly some associated object attributes such as Name etc. An intersection of the keyword lists with appropriate weighting factors as defined by the query VOPD, will provide the matching result. A vocabulary of synonyms with closeness measure can be incorporated into the system to account for differences between words in the index and words in the query.

- **Structure Matching** - the graph characteristics of the VOPD is exploited when comparing two VOPDs. The objects in the VOPD are connected by structural links, notably aggregation, characterization and general (labeled) links. Thus a VOPD can be viewed as a (directed and labeled) graph, where the objects are the vertices and the structural links are the graph edges. Structure Matching is performed between two VOPDs by comparing the underlying graph structures using graph matching techniques [17]. The vertices and links may be considered as labeled, thereby restricting the graph matching to labeled graph comparisons. Again, the choice of labeled vs. unlabeled approaches depend on the parameters and priorities set by the query VOPD, as discussed below.
- **Attribute Matching** - the objects in the VOPD are associated with local attributes (such as color, size, texture, etc.), and global attributes (such as image name, scene illumination, mood, etc.). According to parameters of the query VOPD, these attributes can serve as the basis for the VOPD matching. This matching mode incorporates the low-level, feature based retrieval approaches, used in previously suggested image retrieval methods, into our semantic content-based approach.

More than one mode of matching can, and often should be used during the VOPD matching process. The query VOPD is the determining factor for the mode and method of VOPD matching. Associated with the query VOPD is a set of parameters determining the priorities and weights of the components of the desired image. For example, a query VOPD may describe a preference for a grass background regardless of foreground, or a child smiling regardless of gender, or a sunset image regardless of the scene content or color distribution. A user may wish to retrieve an image of a cat and dog, but will accept an image containing only a dog and regardless of whether it is in the foreground or background. Additionally, the mode of matching can be specifically determined by the user.

The Query VOPD can be considered as a sub-graph of the desired image. The matching modes take this into account by searching for subsets of keywords in the Unstructured Matching mode and by considering sub-graph matching in the Structured Matching mode.

The result of each match mode is a graded measure of similarity, thus in the Unstructured Matching mode, the "distance" between the keyword lists can be measured using a Hebbian type metric (the number of common entities). In the Structured Matching mode, a measure of graph similarity must be defined and used. Using the Attribute Matching, a metric for each measure, such as a color distance metric for the color attribute and a scalar metric for the size attribute, is used.

The final matching result between two VOPDs may be represented as a single scalar value, defining the “distance” between the VOPDs, or as a vector, with each entry corresponding to some aspect of the similarity, i.e., the similarity measure obtained for each matching mode used in the comparison.

9 Work in Progress

The Content Based Image Retrieval project, currently under way, concentrates on developing the two main processes of the system. We assume that the extraction of visual components and the VOPD generation, are performed manually. For a restricted set of images, such as those of a block world, the visual component extraction and the VOPD creation can be automated. At present we focus on two issues: defining rules for VOPD generation and developing specialized algorithms for VOPD matching.

Following the establishment of VOPD generation rules and the matching algorithms, we will focus on the design and implementation of the specialized version of OPCAT which will serve for both index and query VOPD generation and as interface for the retrieval and display of images from the database.

References

1. J. Bigun, S.K. Bhattacharjee, and S. Michel. Orientation radiograms for image retrieval: an alternative to segmentation. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 3, pages 346–350, 1996.
2. A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997.
3. A. Finkelstein C.E. Jacobs and D.H. Salesin. Fast multiresolution image querying. In *Computer Graphics Proceedings - SIGGRAPH 95*, pages p. 277–286, 1995.
4. D. Dori. Arc segmentation in the machine drawing understanding environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1057–1068, 1995.
5. D. Dori. Representing pattern recognition-embedded systems through object-process diagrams: the case of the machine drawing understanding system. *Pattern Recognition Letters*, 16(4):377–384, 1995.
6. D. Dori. Unifying system structure and behaviour through object-process analysis. *Journal of Logic and Computation*, 5(2):227–249, 1995.
7. D. Dori. Analysis and representation of the image understanding environment using the object-process methodology. *Journal of Object Oriented Programming*, September, 1996.
8. D. Dori. Expressing structural relations among dimension-set components using the object-process methodology. *Report on Object Analysis and Design*, 2(6):20–24, 1996.
9. D. Dori. Object-process analysis of computer integrated manufacturing documentation and inspection. *International Journal of Computer Integrated Manufacturing*, 9(5):339–353, 1996.

10. D. Dori. Unifying system structure and behaviour through object-process analysis. *Journal of Object-Oriented Analysis*, July-August:66-73, 1996.
11. D. Dori and M. Goodman. From object-process analysis to object-process design. *Annals of Software Engineering*, 2, 1996.
12. D. Dori and M. Goodman. On bridging the analysis-design and structure-behavior grand canyons with object paradigms. *Report on Object Analysis and Design*, 2(5):25-35, 1996.
13. D. Dori and M. Weiss. A scheme for 3d object reconstruction from dimensioned orthographic views. *Engineering Applications of Artificial Intelligence*, 9(1):53-64, 1996.
14. D.A. Forsyth et. al. Finding pictures of objects in large collections of images. In *Proceedings of the International Workshop on Object Representation in Computer Vision II, ECCV-96*, pages 335-360, 1996.
15. J. Hafner et. al. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729-735, 1995.
16. T. Kato et. al. A sketch retrieval method for full color image database-query by visual example. In *Proceedings. 11th IAPR International Conference on Pattern Recognition*, pages 530-533, 1992.
17. S. Even. *Graph algorithms*. Computer Science Press, Potomac, Md, 1979.
18. C. Faloutsos. Access methods for text. *ACM Computing Survey*, 1:49-74, 1985.
19. G.L. Gimel'farb and A.K. Jain. 9. *Pattern Recognition*, 29:1461-1483, 1996.
20. A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(9):1233-1244, 1996.
21. A. Kankanhalli, J.Z. Hong, and Y.L. Chien. Using texture for image retrieval. In *Proceedings of The Third International Conference on Automation, Robotics and Computer Vision*, volume 3, pages 935-939, 1994.
22. H.C. Lin, L.L. Wang, and S.N. Yang. Color image retrieval based on hidden markov models. *IEEE Transactions on Image Processing*, 6(2):332-339, 1997.
23. W.Y. Ma and B.S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35-51, 1996.
24. M. De Marsicoi, L. Cinque, and s. Levaldi. Indexing pictorial documents by their content: a survey of current techniques. *Image and Vision Computing*, 15(2):119-141, 1997.
25. D. Meyersdorf and D. Dori. The r&d universe and its feedback cycles: an object-process analysis. *R&D Management*, To Appear.
26. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In *SPIE Conference on Storage nad Retrieval for Image and Video Databases*, volume 1908, pages 173-187, 1993.
27. A. Ono, M. Amano, M. Hakaridani, T. Satou, and M. Sakauchi. A flexible content-based image retrieval system with combined scene description keyword. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 201-208, 1996.
28. G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *Proceeding. Third IEEE Workshop on Applications of Computer Vision*, pages 96-102, 1996.
29. A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *Int. Journal of Computer Vision*, 18(3):233-254, 1996.

30. R.W. Picard. A society of models for video and image libraries. *IBM Systems Journal*, 35(3-4):292-312, 1996.
31. Z. Qing-Long, C. Shi-Kuo, and S.S-T. Yan. Iconic indexing and maintenance of spatial relationships in image databases. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2916, pages 385-396, 1996.
32. E. Remias, G. Sheikholeslami, and A. Zhang. Block-oriented image decomposition and retrieval in image database systems. In *Proceedings. International Workshop on Multi-Media Database Management Systems*, pages 85-92, 1996.
33. M. Shneier and M. Abdel-Mottaleb. Exploiting the jpeg compression scheme for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):849-853, 1996.
34. H.G. Stark. On image retrieval with wavelets. *International Journal of Imaging Systems and Technology*, 7(3):200-210, 1996.
35. M.J. Swain and D.H. Ballard. Color indexing. *Int. Journal of Computer Vision*, 7(1):11-32, 1991.
36. D.L. Swets and J.J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831-836, 1996.
37. C.W. Tzong and C.C. Chang. Application of geometric hashing to iconic database retrieval. *Pattern Recognition Letters*, 15(9):871-876, 1994.
38. A. Vailaya, Z. Yu, and A.K. Jain. A hierarchical system for efficient image retrieval. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 356-360, 1996.
39. X. Wan and C.-C.J. Kuo. Image retrieval with multiresolution color space quantization. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2898, pages 148-159, 1996.
40. X. Wan and C.J. Kuo. Image retrieval based on jpeg compressed data. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2916, pages 104-115, 1996.
41. G. Yihong, C.H. Chuan, and Z. Guo. Image indexing and retrieval based on color histograms. *Multimedia Tools and Applications*, 2(2):133-156, 1996.
42. J. You, H. Shan, and H.A. Cohen. An efficient parallel texture classification for image retrieval. In *Proceedings. Advances in Parallel and Distributed Computing*, pages 18-25, 1997.