An Approach to Image Retrieval for Image Databases

T. Gevers and A.W.M. Smeulders

Faculty of Mathematics & Computer Science, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

E-mail: gevers@fwi.uva.nl

Abstract

In this paper, a method is discussed to store and retrieve images efficiently from an image database on the basis of the data structure called E() representation. The E() representation is a spatial knowledge representation preserving the spatial information between objects embedded in symbolic images as an iconic index for the purpose of efficient image retrieval.
The image retrieval method is invariant under, at least, the affine transformation (i.e. translation, rotation and scale) and is able to deal with substantial object occlusion. A metric is defined to express similarity between symbolic images.
Initial experiments carried out for two applications show that the image retrieval method is very efficient and robust to similarity retrieval in image databases. Together with the inherent high parallelism, it makes the method a promising image retrieval method.
keywords: image database, image indexing, similarity retrieval, spatial relations, E() representation, metric, spatial query language.

1 Introduction

Research in the design of image database systems has increased significantly in the last decade [3], [12], [21]. Much attention has been paid to the problems of how to store images and to retrieve them efficiently from an image database. Many data structures have been proposed at the level of pixels, such as pixel-based [7], R-trees [13], quadtrees [20] together with their spatial processing operations. Most of the image database systems combines these spatial processing capabilities with DBMS capabilities for the purpose of storage and retrieval of complex data (e.g. range queries). Other image database systems are still based on the paradigm to store a key-word description of the image content, created by some user on input, in a relational DBMS in addition to a pointer to the raw image data. Image retrieval is then based on the relational DBMS capabilities. However, another approach is required if we consider the wish to retrieve images on the basis of objects and their spatial relations. To that end, an object-oriented data structure is required to represent images preserving information about the objects and their spatial relation in a robust way to enable efficient image indexing and retrieval [4],[5],[19].

In this paper, a method is presented to retrieve images based on the E() indexing scheme. The E() representation preserves the spatial information between objects, embedded in images, as an index. The image retrieval method is invariant under, at least, the affine transformation (i.e. translation, rotation and scale) and is able to deal with substantial object occlusion. A query language is discussed to enable the formulation of a query in terms of objects and their spatial relations. Because the query as well as every image in the image database are represented by an E() representation, image retrieval is reduced to E() representation matching. A metric is given to express similarity between two E() representations. For the purpose of image retrieval, the metric can be used to compare a picture query with each E() representation of every image in database and order the images by their proximity to the query. Images with a low metric are considered the same or similar to the query.

This paper is organized as follows.
Section 2 will start with the introduction of the E() representation scheme for image indexing illustrated by an example. In Section 3, a comparison scheme is presented to retrieve images on the basis of directional relations between object pairs followed by the time and space complexity analysis. Section 4 will discuss another comparison scheme for the purpose of image retrieval based on the sequence in which objects are situated around each other. A spatial query language, suitable for the method, is discussed in Section 5. In Section 6, applications and experiments are presented to illustrate the nature of the method and how the method can be applied to real-world problems. Finally, a summary will be given.

2 Image Indexing: the E() Representation

Retrieval of images is one of the most important issues of image database design. In this paper, we approach this problem on the basis of the observation that images can be discriminated by the presence of objects and their spatial relations. Therefore, a query should be defined in terms of objects and their spatial relations. A typical query is for example "retrieve all images with a car left to a house". Image retrieval is then the process to compute to what extent the images in the database, described by objects and their spatial relations, correspond to the query. To realize the image retrieval process, a data structure is needed to represent the images. The data structure must capture the information about the objects as well as the spatial information in images in a robust way to enable efficient image indexing and retrieval. To that end, a representation scheme is discussed in this section.

The architecture of the image database system consists of a physical image store and a logical database. A symbolic description is obtained for every raw digital image in the physical image store by applying image processing and pattern recognition techniques or by manually tracing outlines of objects. The symbolic description of an image includes the identity of objects and their position in image space. In this paper we make no distinction between low-level objects such as corners, points of inflection and t-junctions [18] or high-level objects such as a rectangular box, a chair, a table etc. We assume that objects have already been recognized automatically or interactively and unique symbolic labels have been assigned to them, yielding a symbolic picture.

More precisely, a symbolic picture f is defined as a mapping Mx$N \rightarrow S$, where $M = (1, 2, ..., m)$ and $N = (1, 2, ..., n)$. S is the power set of V, where V is a set of symbols called the vocabulary. Each symbol represents an object.

A data structure is required to efficiently store symbolic pictures for the purpose of flexible image indexing and retrieval. The data structure should be object-oriented and should be able to preserve the spatial relations between the objects in an elegant and flexible manner. Various different spatial relations have been proposed in literature [1], [15]. The most typical spatial relations are topological relations, describing (partly) overlapping objects, adjacency relations, describing to what extent objects are touching each other, and directional relations such as north, east, south and west. In this paper, we concentrate on directional relations.

From now on, we consider point objects which give an unambiguous account of directional relations. For objects larger then points, we consider their centroids as indicators.

An intuitive understanding of how a symbolic picture f of point objects is represented, is given in the following example. Consider the symbolic picture shown in fig. 1.

The object embedding induced by the symbolic picture can be defined by the combination of all directional relations between each object pair. Thus the overall configuration of objects, also called the gestalt of the image, can be described by the set of binary directional relations. To determine the directional relations between each object pair is to regard each of the objects as the origin of a coordinate system and recording the locations of all other objects. The coordinate system plane may be quantized at different resolutions, making the directional relations between objects as precise or fuzzy as apt for the domain at hand. If simple directional relations north-east, south-east, south-west and north-west are considered, it is sufficient to quantized the coordinate space in four quadrants.

Consider the symbolic picture of fig. 1.a. First, we place object a at the origin of a rectangular coordinate system where the direction of the x-axis and y-axis corresponds to that of the symbolic picture, as shown in fig. 1.b. All other objects are lying in one of the four quadrants. Let's record in which of the four quadrants each of the objects is lying. Object b will be recorded as lying east of object a along the x-axis and north along the y-axis. The two dimensional directional relation north-east is recorded for object b with respect to object a and will be represented as the string $a :>> b$. Further, object c is also north-east with respect to a, yielding the following string $a :>> b >> c$. Finally, we obtain the string $a :>> b >> c >< d$, where the

Figure 1: a: symbolic picture b: object a placed at the origin of a rectangular coordinate system

substring $a:$ denotes that the following substring records how the other objects b, c and d are positioned with respect to a. Similarly, this process is repeated for the other objects as being placed at the origin of the coordinate system, yielding the strings $b :<< a >> c >< d$, $c :<< a << b << d$ and $d :<> a <> b >> c$. The entire set of strings, denoted by $E(a,b,c,d)$, is

$$E(a,b,c,d) = \begin{array}{ll} \{ & (a :>> b >> c >< d) \\ & (b :<< a >> c >< d) \\ & (c :<< a << b << d) \\ & (d :<> a <> b >> c) \quad \} \end{array} \tag{2.1}$$

This coding scheme is inefficient, because the spatial information between any object pair is recorded twice (e.g. north-east(x,y) implies south-west(y,x)). Therefore, the following set of strings is necessary and sufficient to represent the symbolic picture f:

$$E(a,b,c,d) = \begin{array}{ll} \{ & (a :>> b >> c >< d) \\ & (b :>> c >< d) \\ & (c :<< d) \quad \} \end{array} \tag{2.2}$$

A more formal definition of $E()$ is as follows. Let V be a set of symbols. Each symbol represents an object. Let O be a set of spatial operators which specify spatial relations among objects, then

$$E(v_1, v_2, ..., v_n) = \begin{array}{ll} \{ & (v_1 : o_{12}v_2o_{13}v_3...o_{1n}v_n) \\ & (v_2 : o_{23}v_3o_{24}v_4...o_{2n}v_n) \\ & . \\ & . \\ & . \\ & (v_{(n-1)} : o_{(n-1)n}v_n)\} \end{array} \tag{2.3}$$

where $v_1, v_2, ...v_n$ are symbols in V and $o_{ij} \in O$ for $i < j \leq n$. For the example above, the vocabulary is $V = \{a, b, c, d\}$ and $O = \{>>, ><, <>, <<\}$ is the set of simple directional relations:

$$E(v_1, v_2, ..., v_4) = \begin{array}{ll} \{ & (v_1 : o_{12}v_2o_{13}v_3o_{14}v_4) \\ & (v_2 : o_{23}v_3o_{24}v_4) \\ & (v_3 : o_{34}v_4) \quad \} \end{array} \tag{2.4}$$

where $v1v2v3v4$ is $abcd$ and $o_{12} = \{>>\}$, $o_{13} = \{>>\}$,$o_{14} = \{><\}$,$o_{23} = \{>>\}$, $o_{24} = \{<>\}$ and $o_{34} = \{<<\}$. The spatial relations $\{>>\}, \{><\}, \{<>\}$ and $\{<<\}$ denote the directional relations north-east, south-east, south-west and north-west.

The $E()$ representation preserves all simple directional relations among objects induced by the symbolic picture f. In the example, the set of spatial relations, O, is the set of simple directional relations. This set can be extended to include all possible geometrical relations between each two point objects by considering their coordinates (i.e. each point object can be identified by an ordered pair (x, y) of real numbers corresponding to the coordinates in the coordinate system generated by placing a particular object at the origin) as their relation. In fact, o_{ij}, reports how object j is related to object i. For spatial relations, coordinates are used to represent the relation. The set of spatial relations, O, can also be extended to include the sets of topological and adjacency relations as well.

To perform image retrieval, each symbolic image for every image in the image database is translated to an $E()$ representation as well as the query. The query can be specified by drawing an iconic picture on the screen. The iconic picture consists of icons representing objects and their spatial relationships, similar to a symbolic picture, see Section 5. After the user has specified the query graphically, the query is converted into an $E()$ representation. The problem of image retrieval then becomes the matching of two $E()$'s.

3 Image Retrieval by E() Matching

A metric between two $E()$'s is necessary to express to what extent two images, represented by E_1 and E_2 respectively, are similar. If two object embeddings with different sets of objects have to be compared, only those objects which are common to both are considered to contribute to the metric. The advantage to take objects which are common to both embeddings to compute the metric, is that the method is able to deal with substantial occlusion of objects. This is desirable for general image retrieval.

Let two $E()$ representations $E_1(v_1, v_2, ..., v_n)$ and $E_2(v_1, v_2, ..., v_n)$ contain the same set of n distinct objects $V = \{v_1, v_2, ..., v_n\}, v_i \neq v_j, \forall i \neq j \leq n$. Further, let O_1 and O_2 denote the sets of spatial operators of respectively E_1 and E_2. Then the distance is defined as follows:

$$d(E_1, E_2) = \sum_{i=1}^{i=n} \sum_{j=i}^{i=n} w_{ij} f(o_{ij}, p_{ij}), o_{ij} \in O_1, p_{ij} \in O_2 \tag{3.1}$$

where, w_{ij} is a weight factor.
If we use a 2D relation table to store each $E()$ representation, the elements on and below the diagonal of the matrix are zero. The matrix is triangular with total number of elements is $\sum_{i=1}^{i=n}(i-1)$ for n objects. In each entry $E_{ij}, i < j \leq n$ of E(), the coordinate is recorded of object j in the coordinate system generated by placing object i at the origin. The coordinate system is a rectangular coordinate system, usually the Euclidean L_2. In the 2-D case, the coordinate vector $\vec{x} = (x, y) \in R^2$ is stored in each entry expressing the directional relation between object pairs.

Thus, mathematically, we can model the distance between two directional relations as a function $f \in R^2$ of a two dimensional spatial vector $\vec{x} = (x_2 - x_1, y_2 - y_1) = (\Delta x, \Delta y) \in R^2$, where $\vec{x}_1 = (x_1, y_1)$ is the coordinate vector recorded at entry (i, j) for E_1 and $\vec{x}_2 = (x_2, y_2)$ is the vector recorded at the corresponding entry for E_2. For each vector $\vec{x}$, the value $f(\vec{x})$ equals the distance between two directional relations. When $f(\vec{x})$ has been chosen to express the minimum number of directional relations violated to transform relation $\vec{x}_1$ into $\vec{x}_2$, the function f corresponds to the magnitude distance function:

$$f(\vec{x}) = |x| + |y| \tag{3.2}$$

or the Euclidean distance function:

$$g(\vec{x}) = \sqrt{x^2 + y^2} \tag{3.3}$$

Let E_1 and E_2 be the symbolic representations of two object embeddings of the same distinct objects. Let $d(E_1, E_2)$ denote the total minimum number of directional relation violations required to transform E_1 into E_2, then $d(E_1, E_2)$ is a metric and implies that for every E_i other $E_1, E_2, ..., E_n$ not equal to E_i can be ordered in their proximity to E_i. For the purpose of image retrieval, E_i can be seen as the representation of a query and $E_1, E_2, ..., E_n$ the representations of the images in the image database. In this way, images are shown on the screen to the user in accordance to their metric value, where images having a low metric are considered the same or similar to the query.

3.1 Complexity Analysis

The $E()$ representation matching process consists of computing the metric for the picture query against every image in the image database and ordering the images in accordance to their proximity to the query. Because the distance of each pair relations can be computed independently at the same time, the method offers a high inherent parallelism, achieving high computational efficiences when implemented on fast purpose hardware for parallel implementations. Parallel implementation issues are not discussed in this paper.

We now analys the complexity of the method as being implemented on a sequential computer. Let n denote the number of distinct objects common to both representations E_1 and E_2. The time complexity of computing $d(E_1, E_2)$ is $O(n(n-1)/2)$:

$$d(E_1, E_2) = \sum_{i=0}^{i=n} \sum_{j=i}^{j=n} w_{ij} f(o_{ij}, p_{ij}) = \sum_{i=0}^{i=n} \sum_{j=i}^{j=n} f(\vec{x}) = \sum_{i=0}^{i=n} \sum_{j=i}^{j=n} 1 = n(n-1)/2 \tag{3.4}$$

Where $f(\vec{x})$ is the function of equation 3.2. Thus the $E()$ matching time depends on the number n, the number of distinct objects which are common to both $E()$'s, and increases quadraticly with n. Fast and robust correspondence search is presented in [11].

Because a 2D relation table is used to store each $E()$ representation, the elements on and below the main diagonal of the square matrix are zero. Hence, the space complexity of the algorithm is $O(n(n-1)/2)$.

3.2 Invariance under the Affine Transformation

Till now, the method is invariant under translation. For general image retrieval, the method should be invariant, at least, under the affine transformation such that object embeddings can be recognized even if they are transformed (rotated, translated, scaled and sheared).

We first extend the method to be invariant under rotation. An rotation transformation is uniquely defined by the transformation of two non-collinear objects [17]. Therefore, each combination of two non-collinear objects can be chosen to represent all other objects. Specifically, let p_0 and p_1 be two non-collinear objects extracted from a symbolic picture, then any other object from the same symbolic picture can be represented in the 2D space spanned by the basis (p_0, p_1). The origin is chosen to be the middle of p_0 and p_1, see fig. 2.

Figure 2: Basis for transformation.

Let n be the number of objects. The coordinates are computed for all other $n-2$ objects in the new coordinate frame defined by the basis objects. In this way, the coordinates of all $n-2$ objects undergo a rotational transformation equal to the system and therefore the coordinates remain the same with respect to the basis (p_0, p_1). To compare two object embeddings of two images for the purpose of image retrieval, two non-collinear objects common to both object embeddings are chosen and the coordinates of all other objects are calculated with respect to the basis, yielding two $E()$ representations which can be matched. Due to object occlusion or noise, one or more objects may not be present or may be translated in the images to be compared. Hence, not only one basis pair should be taken, but each ordered non-collinear pair of objects must be taken to be the basis of a new coordinate frame. When n is the number of distinct objects, then there exists a maximum of n^2 different basis pairs, yielding n^2 $E()$ representations. Because the time complexity to match two E()'s is $O(n^2)$, the time complexity of the method to be translation and rotation invariant is $O(n^4)$, because n^2 $E()$ representations must be matched, each with complexity $O(n^2)$ in order to compute the best translation- rotation invariant match.

To allow the method to be invariant under the affine transformation (translation, rotation, scale and sheare), a three object basis is required. Let p_0, p_1 and p_2 be three non-collinear objects in 2D extracted from a symbolic picture, and $(p_1 - p_0, p_2 - p_0)$ is the basis with p_0 as the origin spanning the 2D space. The maximum number of basis triplets is n^3 to represent the other $n-3$ objects. For each ordered non-collinear triplet an $E()$ representation is generated. Hence, the number of $E()$'s for an arbitrary image with n objects is n^3 and the time complexity to match two images, both having n identical distinct objects and required to be invariant under the affine transformation, is $O(n^5)$. The great advantage of this approach is that it suitable not only for the affine 2-D case, but also to other useful transformations in 2-D and 3-D, as required by the application, by simply changing the number of basis objects. For example, 4 basis objects is needed for a projective transformation [8]. Because only the number of objects to be taken as a basis for a new coordinate frame differ, it is important to know to what extent the time complexity is affected by the number of basis objects.

More specifically, let k be the number of basis objects needed for the desired transformation. All other $n-k$ objects have transformation invariant coordinates with respect to the k-tuple basis. Then the time and space complexity of the method is $O(n^2)$, $k=1$, and $O(n^{k+2})$, $k>1$.

Assuming that n^k-dimensional $E()$ representation is in the main memory and the function of equation 3.2 is used to calculate the distance of two directional relations, the method consists of very simple basic routines: fetching the coordinates, calculating their distance and accumulate all the distance values. Because these simple basic routines can be computed very fast, the image retrieval method is very efficient and is able to be executed at high speeds allowing on-line real-time image retrieval for a large image database even for complex transformations and large n.

4 Circular Coordinate System

Till so far, the coordinate system is a rectangular coordinate system (i.e. the Cartesian plane), usually the Euclidean L_2, which may be quantized at different resolutions.

Another tessellation of the plane is obtained by regarding the space around the origin as being circular, corresponding to the polar coordinate system without magnitude information, see fig. 3.

The bins are equally sized and for $\theta = \frac{2\pi}{2^n}, n = 2$, the simple directional relations are represented. As opposed to the rectangular system, each location consists of only one coordinate. For $n > 2$, the resolution of the coordinate system increases. For large n, relations are more precisely described.

The interval of natural numbers corresponding to the bins is $(0, 2^n]$, $n \geq 2$. The range of the interval equals the number of bins the circular coordinate system consists of. In this way, the location (i.e. coordinate) of each object is represented by a natural number $\in (0, 2^n]$ corresponding to the bin the object is lying in.

The $E()$ representation is generated by placing each object at the center of the

Figure 3: Circular coordinate system.

circular coordinate system and all object coordinates are recorded and stored as an entry. Let $\vec{x}_1 = (x_1) \in N^+$ be the one-dimensional vector of entry E_{ij} of $E_1()$ describing in which bin object j lands with respect to object i placed at the origin of the circular coordinate system. $\vec{x}_2 = (x_2) \in N^+$ is the one-dimensional vector of entry E_{ij} of $E_2()$. Then $\vec{x} = (x_2 - x_1)$ and $f(\vec{x}) \in N^+$ is defined as:

$$h(\vec{x}) = |x| \qquad (4.1)$$

and the distance $d(E_1, E_2)$ is defined as:

$$d(E_1, E_2) = \sum_{i=1}^{i=n} \sum_{j=i}^{i=n} w_{ij} h(o_{ij}, p_{ij}), o_{ij} \in O_1, p_{ij} \in O_2 \qquad (4.2)$$

where O_1 and O_2 denote the sets of spatial operators of respectively E_1 and E_2.

The method is already translation and scale (i.e. the magnitude of the polar coordinate system is discarded) invariant. To allow the matching of two $E()$'s to be also rotation transformation invariant, two non-collinear objects are taken as a basis. All other objects are represented by the new coordinate frame defined by the basis.

4.1 Matching E()'s on the Basis of Object Sequence Information

The major difference between the rectangular coordinate system and the circular one, is that the latter offers a nice framework which describes the sequence in which objects are positioned around each other.

In this section, we discuss another comparison scheme, based on the circular coordinate system, to match E() representations. The paradigm of the comparison scheme is that the gestalt of an image is described by the sequence in which objects occur around each object. This information is preserved in the $E()$ representation generated from a circular coordinate system, see fig. 4.

Figure 4: Example for n=3. Left: symbolic picture f. Right: symbolic picture g.

For example, the sequence of all objects around a for the symbolic picture f, see fig. 4, is denoted by the string $a : d_1 c_3 b_7$. Because the relative ordering of the objects in the sequence is important, we consider the relative string $a : d_1 c_2 b_3$, where $a :$ denotes that the following substring describes the relative ordered sequence of objects in which they occur by scanning the space around a in a counterclockwise manner. In a similar way, the set of strings for b, c and d are obtained for f :

$$
\begin{array}{ll}
\{ & (a : d_1 c_2 b_3) \\
 & (b : d_1 a_2 c_2) \\
 & (c : a_1 b_1 d_2) \\
 & (d : c_1 a_2 b_3) \quad \}
\end{array}
\tag{4.3}
$$

and for g:

$$\begin{array}{ll} \{ & (a : c_1 d_2 b_3) \\ & (b : c_1 a_2 d_2) \\ & (c : d_1 a_2 b_3) \\ & (d : a_1 b_1 c_2) \quad \} \end{array} \tag{4.4}$$

The sequence in which objects occur with respect to an object is easily derived from an E() representation, because the one-dimensional coordinates of objects coming from the same row, is a natural number corresponding to a bin. Hence, the entries E_{ij} for row i containing the lowest coordinate value, is the first object j detected if the space around object i is scanned in a counterclockwise manner.

For the purpose of image retrieval, two $E()$'s are declared to be the same, if each object in both $E()$'s has the same set of objects situated in the same sequence around it. In order to compute the distance between two $E()$'s, the indexes of the objects of $E_g()$ have to be changed to correspond to the indexes of the objects in $E_f()$.

The distance between two $E()$'s is defined as the minimum of total number of binary exchanges to transform each string of $E_g()$ into an ordered sequence. A binary exchange is defined as the exchange of the position of two objects in the sequence of objects. Notice that circular shifts are not counted.

More formally, consider the string of objects $(O_1, O_2, ..., O_n)$. Each object, O_i, has key value K_i. The key values correspond to natural numbers corresponding to the bins (i.e. coordinates). Therefore, it is obvious that for any two key values K_i and K_j either $K_i = K_j$, $K_i < K_j$ or $K_i > K_j$ (i.e. there exist a transitive ordering relation on the key values). The problem is then to find a permutation, σ, such that $K_{\sigma(i)} \leq K_{\sigma(i+1)}$, $1 \leq i \leq (n-1)$, to obtain the ordering $(O_{\sigma(1)}, O_{\sigma(2)}, ..., O_{\sigma(n)})$. Notice that key values may be identical (i.e. two objects are lying in the same bin), and therefore the above defined, σ, is not unique. During the computation of permutation, σ, the minimum number of binary exchanges to transform $(O_1, O_2, ..., O_n)$ into the desired ordering $(O_{\sigma(1)}, O_{\sigma(2)}, ...O_{\sigma(n)})$ is calculated. Let $d(E_1, E_2)$ denotes the minimum number of total binary exchanges required to transform E_1 into E_2, then $d(E_1, E_2)$ is a metric and can be used to compare two $E()$'s.

5 Spatial Query Language

The purpose of a spatial query language is to enable the user to formulate queries in an intuitive and flexible manner involving both spatial and non-spatial predicates. A review and comparison of existing query languages can be found in [9]. For the image retrieval method, a spatial query language is required to enable the formulation of a query consisting of objects and their spatial relations. An intuitive way to achieve this, is to specify queries graphically by generating an iconic picture on the screen. In this way, the query-by-example paradigm and the concept of icon-oriented visual interface are combined, yielding a powerful spatial query language suitable for the method. After the query has been formed, the query, which is graphically described by an iconic picture similar to symbolic picture, is translated into its $E()$ representation. Then the non-spatial information of the objects is retrieved and the corresponding $E()$ representation is selected for each image in the image database based on standard DBMS capabilities. Image retrieval is then transformed to the problem of $E()$ matching, where the $E()$ representation of the query is compared to the E() representation of every image in the image database. The images are ordered in their proximity to the query. The following techniques can be used to formulate a query.

5.1 Query by Example Image

This technique is based on the query-by-example-image paradigm, which means that images are retrieved on the basis of an example image. The example image is selected by the user at run time and corresponds to an image in the image database. The selected image is represented by its symbolic picture. The user is allowed to specify the following image retrieval parameters:

- 1. The coordinate system (i.e. rectangular or circular).
- 2. Resolution of the coordinate system.
- 3. The distance function between the objects.
- 4. Comparison schema.
- 5. Viewing transformation (i.e. translation, rotation, scale and so on).

Then the $E()$ representation is generated by deducing the spatial relations between the objects, induced by the symbolic picture of the query and the retrieval parameter settings. Because every image in the image database is properly described by its $E()$ representation, image retrieval is reduced to $E()$ matching. The metric is used to order the images with respect to the example image.

The advantage of specifying a picture query based on the query-by- example-image paradigm, is that images in the image database are classified with respect to an example image in an intuitive and simple manner which can be useful in various applications. However, it is also very likely that no example image is available which expresses precisely the structural model the user is interested in. Therefore, the query can also be specified graphically by formulating an iconic picture on the screen.

5.2 Query by Iconic Picture

A large set of icons is available to the user. Each icon represents an object. To each icon a set of attributes is assigned. The user can select an icon and set the desired attribute values. For example, the user can select an icon representing a city with the attribute "*population*" attached to it. Once the desired icons (i.e. objects) are selected and their attributes are properly set, the user is allowed to move the icons around and place them properly on the screen with respect to their spatial relations. In this way, the non-spatial predicates are specified by the selection of icons and their attributes and the spatial predicates are expressed by the locations of the icons on the screen with respect to each other. The user is allowed to specify the image retrieval parameters. The $E()$ representation of the iconic picture query is generated by deducing the spatial relations induced by the iconic picture query and the image retrieval parameter settings. Then the non-spatial information of the objects is retrieved for each image and their corresponding $E()$ representation is matched against the query. The images are ordered in accordance to their metric and shown on the screen.

5.3 Query by Iconic Example

The difference between this mode and the query-by-iconic-picture mode is that it does not matter how the icons are located with respect to each other on the screen, because the user has to specify the spatial relations between the icons explicitly. After the desired icons have been selected and their attributes are properly set, the user may specify zero or more relations, in addition to a distance function for directional relations, between each two objects. To allow the specification of distance functions between each two icons, queries can be constructed having partial embeddings which may be precisely defined and other partial embeddings more fuzzy. The goal of enabling more then one relations to be specified between each two icons is to retrieve images in which the relation between two objects may differ. More then one relation to be specified between two the same icons, is interpreted by the or-form. For example, picture queries may be formulated to retrieve images with "a car to the east or to the south of a house".

After the retrieval parameters have been set for each two icons in the query, the E() representation is generated and matched with those of the images in the image database.

6 Applications & Experiments

The image indexing and retrieval method has been implemented and integrated as a part of a larger prototype information system [10]. The purpose of the information system is that it is used as a research vehicle to investigate important issues with regard to image database design. The information system has been implemented in C in combination with the X widget set, on a SUN-SPARC workstation with UNIX as operating system. The ScilImage package, [16], provides the image processing functionality.

In this section, applications are discussed to illustrate the nature of the method and its importance in real-world situations. We discusses both image database and computer vision problems, that need a means of comparing images, represented by their object embeddings, to their benefit.

6.1 Object Recognition

The method can be used for model-based recognition of rigid and deformable 2-D and 3-D objects under different viewing transformations in cluttered scenes and is able to deal with substantial occlusion.

Object recognition is an important problem for image databases as well as for computer vision. The model-based approach to object recognition, in which a predefined model of an object is matched with an input image, has been proved very useful[2],[6]. In the sequel, we consider flat object recognition which is the case when the object is orthogonally projected on the retina (i.e. viewing plane). In this way, 2-D objects can be recognized from 2-D data. Although the 2-D data might be obtained by different sensors, we assume that the 2-D data is the output of a camera (i.e. images).

The starting point for object recognition, is the way an object can be described. We assume that the object to be recognized can be represented by local geometrical features. Those features may be domain independent, such as corners, high curvature points, T-junctions and so on or domain dependent[10]. Domain dependent features are features from which the object is composed. All of these local geometrical features can be automatically extracted from each image in the image database. The type of features to be extracted from every image depends on the object to be recognized. In general, the features should be specific for the particular object to be recognized and the more different types features are used the more effective the method will be. Because the object is also represented by the chosen features, the problem of object recognition is then to find a match between the image features and the object features. For general object recognition, the match should be invariant under, at least, the affine transformation.

Taking this approach as starting point, the method can be used as follows. First, after extracting the appropriate features from every image in the image database, an E() representation is generated for each image preserving the identity of the features and their geometrical relations as an index. Then, the object modeling step consists of formulating the geometrical model, consisting of features and their spatial relations, as a query to the image database. For example, the user may select an icon representing a local feature such as a corner with the attribute "*angle*" attached to it. After the user has selected the icon and the attributes are properly set, spatial relations may be specified with other icons (i.e. query-by-iconic-example). To allow the specification of distance functions between each two icons, a geometrical object model can be obtained consisting of subsets of features which are defined as precise as necessary enabling the recognition of partly or completed rigid or deformable objects. After the user has specified the retrieval parameters (i.e. coordinate system, resolution, comparison schema and the required viewing transformation), the E() representation of the object model is obtained. Then the method is used to discover a match, consistent with the transformation, of the subset of object features and each compatible subset of the features in every image in the database indexed by their E() representation. The metric information is used to order the candidate objects found in the images. The candidate object with the lowest metric are expected to be the same or similar to the model object.

Tentative experiments have been carried out to test the usefulness and performance of the system. A picture query have been specified to retrieve images containing different electronic components. The query consisted of domain dependent features and their directional relations. The goal was to recognize the different electronic components in a small image database of 20 images of electronic schemes. The system found all images in descending order of quality of resemblance containing the components and relationships as specified by the search request. It took approximately 0.2 seconds per image on a standard SUN-SPARC station to retrieve the non-spatial information of the objects, executing the matching method and ordering the images in descending order of resemblance.

6.2 Similarity Retrieval for Medical Images

The second application discussed in this paper, to present an example of how the method can be applied to real-world problems, is similarity retrieval in medical image databases. The problem at hand is to retrieve images from a medical image database, containing MR images of the heart, that have the same tomographic sections. Retrieval of images with similar tomographic sections is a very common problem in cardiac image databases [14]. Is is known that the embeddings of chambers and blood vessels are important indicators to determine the tomographic section of a cardiac image. It is desirable for the radiologist to compare an example image at hand with images taken from the same tomographic section in order to establish a more reliable diagnosis. Therefore, the radiologist should be able to retrieve images from a medical image database with the same tomographic section in an intuitive and flexible way for comparison. In the context of the method, the user is able to formulate the desired tomographic section by means of the geometrical model, consisting of most informative objects (i.e. called parts in medical terms) and of course their spatial relations. The choice of parts that have to be present in the tomographic section model is made by the user. For example, the most relevant parts of a tomographic section of a MR image of the heart are: left ventricle, right ventricle, left atrium, right atrium, ascending aorta and the pulmonary artery. At run time, the user selects the icons representing these parts and sets the appropriate attributes. After the user has selected the parts, represented by icons and their attributes, he or she is allowed to place the parts properly with respect to each other determining the spatial relation between them. The retrieval parameters are entered and the E() representation of the tomographic section model is constructed. The E() representation is then matched with those of the images in the image database and ordered in accordance to their metric.

Experiments have been carried out on an image database of more than 200 MRI images taken from the chest containing a variety of planar cross-sections through a large variety of patients. The images have been recorded at the Yale University Medical School facilities. A search request was made which consisted of four icons, representing the parts left ventricle, right ventricle, left atrium and right atrium, and their directional relations. With this search request, it was the aim to find in the database those images which contained the same tomographic section. The result was that 7 out of the first 7 highest ranked images rightfully contained the desired plane.

7 Summary

In this paper, a method is discussed to retrieve images efficiently based on the data structure called E() representation. The method is invariant under, at least, the affine transformation and is able to deal with substantial object occlusion. The E() representation preserves the information about objects and their spatial relations as an index. Because the query as well as every image in the image database can be described by an E() representation, image retrieval is reduced to E() representation matching. E() matching is the process that computes to what extent two E() are similar. To that end, two comparison schemes are discussed. First we consider the paradigm that the overall configuration of objects in an image can be described by the combination of spatial relations between each object pair. The second comparison scheme is based on the information expressed by the sequence in which objects are situated around each other. For both comparison schemes a metric is given. For the purpose of image retrieval, the metric can be used to compare a picture query with each E() representation of every image in database and order the images by their proximity to the query. Images with a low metric are considered the same or similar to the query. A query language is discussed to enable the formulation of a query in terms of objects and their spatial relations. Two applications are discussed, that need a means of comparing images, to illustrate the nature of the image retrieval method in real-world situations. Experiments carried out for the two applications show that the method is very robust and efficient and together with the inherent high parallelism it makes the method a promising retrieval method.

Acknowledgements

The research project is supported by NIH: NLM-Ep(1 R01LM05007-01A1)

References

[1] Ballard D. H. and Brown C. M., **Computer Vision**, Prentice-Hall, 1982.

[2] Besl, P. J. and Jain, R. C., **Three-Dimensional Object Recognition** , ACM Computing Surveys, 17(1), 1985, pp. 75-154.

[3] Chang, S. K., **Principles of Pictorial Information Systems Design**, Prentice-Hall, Englewood Cliffs, NJ.

[4] Chang, S. K., Shi, Q. Y. and Yan, C. W., **Iconic Indexing by 2-D Strings**, IEEE Trans. Pattern Anal. Machine Intell., vol. 9, no. 3, 1987, pp. 413-428.

[5] Chang, S. K. and Jungert, E., **A Spatial Knowledge Structure for Image Information Systems using Symbolic Projections**, Proc. Fall Joint Comp. Conf., Dallas, TX, 1986, pp. 79-86.

[6] Chin, R. T. and Dyer, C. R., **Model-Based Recognition in Robot Vision** , ACM Computing Surveys, 18(1), 1986, pp. 67-108.

[7] Chock, M., Cardenas, A., F. and Klinger, A., **Database Structure and Manipulation Capabilities of the Picture Database Management System (PICDMS)** , IEEE Trans. Pattern Anal. Machine Intell., vol. 6, no. 4, 1984, pp. 484-492.

[8] Delone, B. N. and Raikov, D. A., **Analytic Geometry** , Vol. 2, Moscow, 1949.

[9] Egenhofer, M., **Spatial Query Languages** , PhD Thesis, University of Maine, Orono, 1989.

[10] Gevers, T. and Smeulders A. W. M., **Enigma: An Image Retrieval System** , International Conference on Pattern Recognition, The Hague, The Netherlands, vol. II, 1992, pp. 697-700.

[11] Grimson, W.E.L, **Object Recognition by Computer**, Cambridge, MA: MIT Press, 1990.

[12] Guenther, O. and Buchmann A., **Research Issues in Spatial Databases** , SIGMOD RECORD, vol 19, no. 4, 1990, pp. 61-68.

[13] Guttman, A., **R-trees: a Dynamic Index Structure for Spatial Searching** , Proc. ACM-SIGMOD Int. Conf. Management of Data, June 18-21, 1984, pp. 47-57.

[14] Hemant, D. T., Jaffe, C. C. and Duncan, J. S., **Arrangements: A Spatial Relation Comparing Part Embeddings** , International Conference on Pattern Recognition, The Hague, The Netherlands, vol. I, 1992, pp. 91-94.

[15] Hildreth, E. C. and Ullman, S., **The Computational Study of Vision in Foundations of Cognitive Science**, M.I.T. Press, 1989.

[16] Kate ten, T., Balen van, R., Smeulders, A.W.M., Groen, F.C.A., Boer den, G., **SCILAIM: a Multi-level Interactive Image Processing Environment**, Pattern Recognition Letters 11, 1990, pp. 429-441.

[17] Klein, F., **Elementary Mathematics from an Advanced Standpoint; Geometry**, Macmillan, NY, 1925.

[18] Lamdan, Y, Schwartz. J. T. and Wolfson, H. J., **On Recognition of 3-D Objects from 2-D Images**, Proc. of IEEE Int. Conf. on Robotics and Automation, Philadelphia, Pa., 1988, pp. 1407-1413.

[19] Lee, S. Y. and Hsu, F. J., **Picture Algebra for Spatial Reasoning of Iconic Images represented in 2D C-string**, Pattern Recognition Letters, 12, 1991, pp. 425-435.

[20] Samet, H., **The Quadtree and Related Data Structures**, ACM Computer Surveys, vol. 16. no. 2, 1984, pp. 187-260.

[21] Tamura, H. and Yokota, N., **Image Database Systems: A Survey**, Pattern Recognition, Vol. 17, no. 1, 1984, pp 29-43.