Proceedings of International Conference on Distributed
Systems, Software Engineering and Database Systems

Spatial Reasoning and Similarity Retrieval for Pictorial Databases Indexed by RCOS-strings with Shape Approximation*

P. W. Huang

Department of Applied Mathematics
National Chung-Hsing University
Taichung, Taiwan
huang@flower.amath.nchu.edu.tw

Abstract

Picture indexing and abstraction techniques are very important for pictorial database systems. Previous approaches, including 2D string, 2D G-string, 2D C-string, 2D B-string, and RCOS-string, ignore the shape information about objects and cause remarkable distortion in picture reconstruction, inaccuracy in spatial reasoning, and ambiguity in similarity retrieval. In this paper, we present a split-and-merge method which will use a set of rectangles to approximate an object and adopt RCOS-string as the structure for iconic indexing. The spatial reasoning rules and a similarity retrieval algorithm based on this concept are also described. We built a prototype system and demonstrated that picture reconstruction, spatial reasoning and similarity retrieval become much more accurate than all previous approaches.

1. Introduction

In a pictorial database system, abstractions of pictures are very important for iconic indexing, similarity retrieval, query processing, browsing/visualization and spatial reasoning. There are many ways of abstracting the characteristics from a picture. One way is to construct a corresponding symbolic picture which can be represented by a two-dimensional (2D) string as proposed by Chang *et al.* [1,2]. The basic idea of the 2D string is to abstract the objects of a picture as points and then project these points along the x- and y-coordinates to form two strings representing the relative positions of objects in the x- and y-axis, respectively. The 2D string representation has the advantage of preserving spatial relationships among the objects of a picture and therefore facilitates spatial reasoning. Since a pictorial query can also be specified as a 2D string, the problem of pictorial information retrieval becomes a problem of 2D subsequence matching. This approach thus allows an efficient and natural way to construct iconic indexes for pictures. Sketch pictures can also be quickly reconstructed from such 2D strings for visualization. Thus, the 2D string representation with appropriate picture reconstruction algorithm supports browsing for the pictorial database.

However, many spatial relationships such as "partly overlapped" or "contained" can not be clearly represented in 2D strings if non-zero sized objects are abstracted as points. Therefore, Jungert [3], Chang *et al.* [4], Jungert and Chang [5] extended the idea of 2D strings to form 2D G-strings by introducing several new spatial operators to represent more relative positional relationships among objects of a picture. The 2D G-string representation embeds more information about spatial relationships between objects, thus facilitates spatial reasoning about shapes and relative positions of objects.

Following the same concept, Lee and Hsu [6] proposed the 2D C-string representation based on a special cutting mechanism. Comparing with 2D G-strings, the number of parts generated by this new cutting mechanism is reduced significantly, thus the lengths of the strings representing pictures are much shorter while still preserving the spatial relationships among objects. The 2D C-string representation is more economical in terms of storage space efficiency and navigation complexity in spatial reasoning.

In 2D G-string and 2D C-string representations, objects may be partitioned into subparts and stored in the data structure. The subparts belonging to the same objects must be considered as a whole during the process of spatial reasoning. However, the partition process is quite time-consuming and spatial reasoning based on subparts for answering queries will create a large storage overhead [7]. For solving this problem, Lee *et al.* [8] proposed 2D B-strings to represent pictures without the need of partitioning objects. The RCOS-string proposed by Chang and Lee [7] also avoids the time-consuming partition process. The difference between 2D B-string and RCOS-string is that the former records the sequence order of the boundaries of all objects implicitly while the latter records the relative positions of the boundaries of all objects explicitly. Thus, spatial reasoning based on RCOS-strings can be done more efficiently simply by two decision trees [7].

In all of the above approaches, each object of a picture is abstracted either as a point or as a minimum

*This work was supported by National Science Council, Republic of China, under Grant NSC86-2213-E-005-006

bounded rectangle (MBR). The former cannot capture all spatial relationships between non-zero sized objects. The latter suffers inaccuracy in spatial reasoning and causes ambiguity in similarity retrieval as well as remarkable distortation in picture reconstruction. In this paper, we describe a split-and-merge heuristic method that will use rectangles to approximate non-zero sized objects. Once all the objects of a picture are approximated by a set of rectangles, we then convert the picture into a RCOS-string which in turn will be used as an iconic index for the original picture. The number of rectangles approximating an object can be controlled as less as possible to keep RCOS-string representation short while eliminating the possibility of misjudging the spatial relationship between any two objects. Through a real implementation for our system, we have demonstrated that both of spatial reasoning and similarity retrieval become much more accurate for pictures of arbitrary complexity.

The remainder of this paper is organized as follows. Section 2 reviews 2D B-string and RCOS-string representations. In Section 3, we describe a split-and-merge method to generate approximation rectangles for each object. We also show how to use RCOS-strings to establish iconic indexes for pictures containing approximation rectangles. Section 4 describes the rules for spatial reasoning from which more accurate spatial relationship between any two objects can be determined. Section 5 presents a similarity retrieval algorithm based on the spatial relationships defined in Section 4. Then, the implementation of a prototype system based on our concept is described in Section 6. Finally, concluding remarks are given in the last section.

2. A Review of 2D B-string and RCOS-string

Consider the pictures with non-zero sized objects. Let S be a set of object symbols which can be used to represent the begin- or end-bounds of the projections of objects along the x- or y-axis. The special symbol "=" indicates that the projections of two objects either have the same bound or is at the same location. Then, a 2D B-string, represented as (U_x, U_y), is a pair of 1D strings over S $\cup$ {=} such that U_x and U_y represent the relative sequencing or ordering of the bounds of the object projections along the x- and y-axis, respectively. For example, given the symbolic picture shown in Fig. 1, the corresponding 2D B-string representation is (ABAB=CCDD, AA=CBBDCD). Notice that the first occurrence of an object symbol in U_x (or U_y) denotes the begin-bound of that object and the second occurrence of the same object symbol in the same 1D string denotes the end-bound. Furthermore, the position of a symbol in U_x (or U_y) indicates the begin-rank (or end-rank) of that symbol along the x-axis (or y-axis). For instance, the begin- and end-ranks for object B along the x-axis are 2 and 4, respectively. Since we have B=C in U_x and this symbol B is in its second occurrence while the symbol C is in its first occurrence, we conclude that the begin-rank of C is equal to the end-rank of B, which is 4, in the x-axis. Thus, we can compute the ranks for the boundaries of all objects in both directions from a 2D B-string. From these ranks, we can easily reconstruct an iconic picture for the given 2D B-string.

The concept of RCOS-string is similar to that of 2D B-string except that the RCOS-string records the begin- and end-ranks of all objects explicitly in the string representation. Let S={O_1, O_2, ..., O_N} be the collection of object symbols in a symbolic picture. Also, let b_i^x and b_i^y represent the begin-ranks of object O_i along the x- and y-axis, respectively. Let e_i^x and e_i^y represent the end-ranks of object O_i along the x- and y-axis, respectively. Then, a RCOS-string representation has the form of (O_1 O_2 ... O_N, (b_1^x, b_1^y) (e_1^x, e_1^y), ..., (b_N^x, b_N^y) (e_N^x, e_N^y)). Let us look at the picture shown in Fig. 1, the RCOS-string representation for this symbolic picture is (A B C D, (1, 1) (3, 2), (2, 3) (4, 4), (4, 2) (5, 6), (6, 5) (7, 7)).

Spatial reasoning based on RCOS-strings is very efficient because the begin- and end-ranks of all objects in a picture are known already and they can be fed into a decision tree of depth 3 to determine one of the 13 possible spatial relationships between any two objects in the x-direction (or y-direction), as described in Reference [7]. Thus, the advantage of RCOS-string over 2D B-string is that spatial reasoning is more efficient because there is no need to compute the ranks for the boundaries of objects in RCOS-strings.

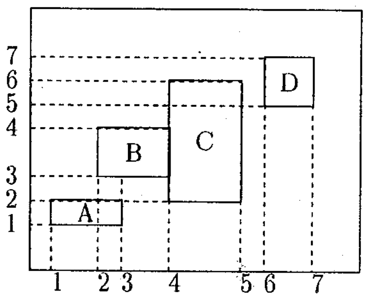

Fig. 1. 2D B-string: (ABAB=CCDD, AA=CBBDCD).
RCOS-string: (ABCD, (1,1)(3,2), (2,3)(4,4), (4,2)(5,6), (6,5)(7,7)).

3. Iconic Indexing by RCOS-string with Shape Approximation

As we mentioned before, all previous approaches except 2D string use an MBR to approximate an object in a picture for iconic indexing. However, instead of using a single MBR, we can use a set of connected rectangles to better approximate the shape of an object by applying a "split-and-merge" method as follows. First, in the splitting phase, we use several horizontal and vertical grid lines to cut the object along the x-axis and y-axis, respectively, so that every object in a picture is partitioned into a group of grids. In the merging phase, a heuristic algorithm is performed to merge these grids into a group of rectangles such that each object can be viewed as a composition of one or more connected rectangles.

The number of grid lines for cutting objects vary from case to case. For simple objects, only a few grid lines may be sufficient to obtain good approximation for them. Thus, increasing the number of grid lines may not improve too much on shape approximation in this case. On the other hand, objects with complex shapes require more grid lines to cut the objects in order to obtain better approximation. However, more grid lines will result in more rectangles for approximating the objects with complex shapes. The larger the number of approximation rectangles in a picture, the longer the RCOS-string representing the picture. Consequently, a longer RCOS-string representation will require more storage space and process time.

Our goal is to keep all RCOS-strings in iconic indexes being as short as possible while still allowing us to judge the spatial relationships between any two objects correctly. Therefore, our strategy is to provide as less grid lines as possible so that misjudgement of spatial relationship between any two objects will never happen.

Notice that the ranks for the boundaries of objects in the x- or y-axis are consecutive positive numbers starting from 1 in the original RCOS-string representation. Actually, the ranks of the begin- and end-bounds of objects can be replaced by the positions of these bounds in terms of the pixel counts in a picture. Such substitutions can be done very easily and have the advantage of causing less distortion in picture reconstruction while still maintaining the relative orders for the bounds of all objects in both of x- and y-directions.

Consider the picture shown in Fig. 2(a), where object A is approximated by three rectangles A_1, A_2, A_3 and object B is approximated by another two rectangles B_1, B_2. The RCOS-string representation for this picture is (A_1 A_2 A_3 B_1 B_2, (10, 25) (30, 60), (30, 10) (60, 80), (60, 10) (90, 50), (70, 70) (110, 100), (110, 80) (130, 90)). The iconic picture reconstructed from this RCOS-string is shown in Fig. 2(b). Notice that if the objects of the picture are represented by their MBR without shape approximation, then the RCOS-string for the this picture will be (A B, (10, 10) (90, 80), (70, 70) (130, 100)). The sketch picture reconstructed from the second RCOS-string is shown in Fig. 2(c). It is obvious that the first RCOS-string has better shape approximation than the second RCOS-string at the expense of five times longer in string-length. However, instead of deriving a "disjoint" spatial relationship between objects A and B from the second RCOS-string, we will derive a "partly overlapped" relation which is incorrect. Since spatial relationships among objects are probably the most important factors that must be considered for spatial reasoning and similarity retrieval, our major concern is to find a short enough RCOS-string to allow us to determine the spatial relationships among objects correctly. Towards this goal, we can simply use two rectangles A_1 and A_2 to approximate object A and use the MBR of object B to approximate object B. The resultant RCOS-string becomes (A_1 A_2 B, (10, 10) (60, 80), (60, 10) (90, 50), (70, 70) (130, 100)), which has a reasonable string-length and still allows us to identify the spatial relationship between object A and object B correctly, as shown in Fig. 2(d).

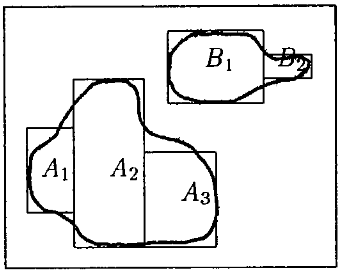

Fig. 2(a). Object A is approximated by three rectangles A_1, A_2, A_3 and object B is approximated by another two rectangles B_1, B_2. The corresponding RCOS-string for the above picture containing approximation rectangles is ($A_1A_2A_3B_1B_2$, (10,25)(30,60), (30,10)(60,80), (60,10)(90,50), (70,70)(110,100), (110,80)(130,90)).

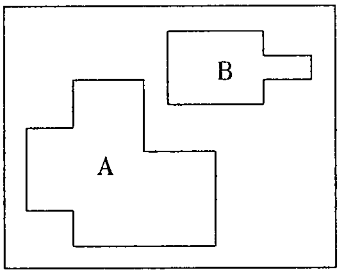

Fig. 2(b). The picture reconstructed from the above RCOS-string.

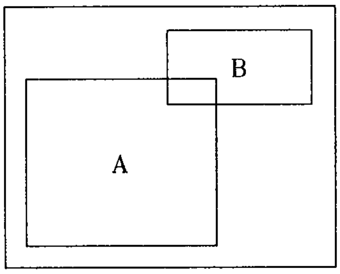

Fig. 2(c). The picture reconstructed from the RCOS-string (AB, (10,10)(90,80), (70,70)(130,100)), where each object is approximated by its MBR.

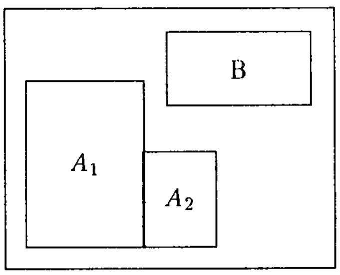

Fig. 2(d). If the picture is indexed by the RCOS-string (A_1 A_2 B, (10,10)(60,80), (60,10)(90,50), (70,70)(130,100)), then the category spatial relationship between A and B can be determined correctly while the string-length is still short.

4. Spatial Reasoning

Let A and B be any two objects in a given picture. Assume that A is approximated by $A_1 \cup A_2 \cup \ldots \cup A_n$ and B is approximated by $B_1 \cup B_2 \cup \ldots \cup B_m$, where A_i and B_j are rectangles for $1 \leq i \leq n$, $1 \leq j \leq m$. Let $(X_{A_i}\ Y_{A_i})$ and $(X'_{A_i}\ Y'_{A_i})$ be the coordinates of the lower-left corner and the upper-right corner of rectangle A_i, respectively. Then, the length of A_i is $L_{A_i} = X'_{A_i} - X_{A_i}$ and the width of A_i is $W_{A_i} = Y'_{A_i} - Y_{A_i}$.

In Reference [6], Lee and Hsu divided all 169 possible spatial relations between any two objects into five categories, namely, DISJOINT, JOIN, PART_OVLP, CONTAIN, and BELONG. Now, we can use the following rules to determine the category spatial relationship between any two object:
C1: "A belongs to B" denoted by (BELONG A B) *iff* $\forall p \in A_i$, $\exists B_j \subset B$ such that $p \in B_j$.
C2: "A contains B" denoted by (CONTAIN A B) *iff* (BELONG B A).
C3: "A partly overlaps with B" denoted by (PART_OVLP A B) *iff* $\exists\ A_i \subset A$, $\exists\ B_j \subset B$ such that the intersections of open intervals

$(X_{A_i}, X_{A_i} + L_{A_i}) \cap (X_{B_j}, X_{B_j} + L_{B_j}) \neq \emptyset$ and
$(Y_{A_i}, Y_{A_i} + W_{A_i}) \cap (Y_{B_j}, Y_{B_j} + W_{B_j}) \neq \emptyset$.

C4: "A separates from B" denoted by (DISJOINT A B) *iff* $\forall\ A_i \subset A$, $\forall\ B_j \subset B$, one of the following three conditions holds:
(a) $(X_{A_i}, X_{A_i} + L_{A_i}) \cap (X_{B_j}, X_{B_j} + L_{B_j}) \neq \emptyset \Rightarrow Y_{A_i} > Y_{B_j} + W_{B_j}$ or $Y_{A_i} + W_{A_i} < Y_{B_j}$
(b) $(Y_{A_i}, Y_{A_i} + W_{A_i}) \cap (Y_{B_j}, Y_{B_j} + W_{B_j}) \neq \emptyset \Rightarrow X_{A_i} > X_{B_j} + L_{B_j}$ or $X_{A_i} + L_{A_i} < X_{B_j}$
(c) $(X_{A_i}, X_{A_i} + L_{A_i}) \cap (X_{B_j}, X_{B_j} + L_{B_j}) = \emptyset$ and
$(Y_{A_i}, Y_{A_i} + W_{A_i}) \cap (Y_{B_j}, Y_{B_j} + W_{B_j}) = \emptyset$.
C5: "A is edge-to-edge connected with B" denoted by (JOIN A B) *iff* None of the above rules can be applied.

The following rules are used to determine the directional relationship between any two objects A and B:
D1: "A is east of B" denoted by (EAST A B) *iff*
$\exists\ A_{i1}, A_{i2} \subset A$ and $\exists\ B_{j1}, B_{j2} \subset B$ such that
$X_{A_{i1}} > X_{B_{j1}}$ and $X_{A_{i2}} + L_{A_{i2}} > X_{B_{j2}} + L_{B_{j2}}$,
where

$X_{A_{i1}} = min\{X_{A_k} \mid \forall A_k \subset A\}$,
$X_{B_{j1}} = min\{X_{B_k} \mid \forall B_k \subset B\}$,
$X_{A_{i2}} + L_{A_{i2}} = max\{X_{A_k} + L_{A_k} \mid \forall A_k \subset A\}$,
$X_{B_{j2}} + L_{B_{j2}} = max\{X_{B_k} + L_{B_k} \mid \forall B_k \subset B\}$.

D2: "A is west of B" denoted by (WEST A B) *iff* (EAST B A).
D3: "A is south of B" denoted by (SOUTH A B) *iff*
$\exists\ A_{i1}, A_{i2} \subset A$ and $\exists\ B_{j1}, B_{j2} \subset B$ such that
$Y_{A_{i1}} < Y_{B_{j1}}$ and $Y_{A_{i2}} + W_{A_{i2}} < Y_{B_{j2}} + W_{B_{j2}}$, where

$Y_{A_{i1}} = min\{Y_{A_k} \mid \forall A_k \subset A\}$,
$Y_{B_{j1}} = min\{Y_{B_k} \mid \forall B_k \subset B\}$,
$Y_{A_{i2}} + W_{A_{i2}} = max\{Y_{A_k} + W_{A_k} \mid \forall A_k \subset A\}$,
$Y_{B_{j2}} + W_{B_{j2}} = max\{Y_{B_k} + W_{B_k} \mid \forall B_k \subset B\}$.

D4: "A is north of B" denoted by (NORTH A B) *iff* (SOUTH B A).
D5: "A is northeast of B" denoted by (NORTHEAST A B) *iff* (NORTH A B) and (EAST A B).
D6: "A is northwest of B" denoted by (NORTHWEST A B) *iff* (NORTH A B) and (WEST A B).
D7: "A is southeast of B" denoted by (SOUTHEAST A B) *iff* (SOUTH A B) and (EAST A B).
D8: "A is southwest of B" denoted by (SOUTHWEST A B) *iff* (SOUTH A B) and (WEST A B).
D9: "A is surrounded by B" denoted by (SURROUNDED_BY A B) *iff* (DISJOINT A B) and at least three of the following relations hold:
(EAST B A), (WEST B A), (SOUTH B A), (NORTH B A).

5. Similarity Retrieval

Similarity retrieval is to retrieve pictures that are similar to the query picture described in a pictorial query [9] supplied by the user. There are many ways of comparing pictures; however, the spatial relationships among objects in a picture are probably the most important factors to be considered for measuring the degree of similarity between any two pictures in many pictorial database systems [1-2, 6-8, 10-14]. Therefore, our similarity retrieval algorithm is also based on this concept.

Let S = {BELONG CONTAIN PART_OVLP DISJOINT JOIN EAST WEST SOUTH NORTH}. The elements in S are called the "basic spatial relations." Given any two objects A and B in a picture f represented by a RCOS-string, we can apply the spatial reasoning rules listed in Section 4 to obtain a set of "basic spatial relations" R=$\{r \mid r \in$ S and $(r\ A\ B)$ holds in $f\}$. Then, R is called the "basic spatial relation set" in a picture f for the ordered pair of objects $(A\ B)$. Also notice that if $A_1, A_2, \ldots, A_n$ appear in a RCOS-string, then we known that each A_i is a constituent part of object A and we call A the "major object symbol" in our convention. Now, we are ready to present our similarity retrieval algorithm as follows.

Algorithm. Similarity retrieval.
Inputs. f_q, the RCOS-string of a query picture; $D=\{f_i \mid 1 \leq i \leq N\}$, where f_i is the RCOS-string of the ith database picture.
Outputs. A list of sequence numbers of database pictures: $Result=\{i_1, i_2, \ldots, i_m\}$, where $1 \leq i_1 < i_2 < \ldots < i_m \leq N$.

1. Generate U, the set of all possible pairs of major object symbols (O_1, O_2) from f_q such that $O_1 < O_2$ in alphabetical order.
2. $i \leftarrow 1$; $Result \leftarrow \emptyset$.
3. For each pair $(O_1\ O_2) \in U$
   (a) If either O_1 or O_2 is not a major object symbol in f_i, then GoTo 5.
   (b) Apply spatial reasoning rules to f_q and generate the basic spatial relation set R_q for object pair $(O_1\ O_2)$.
   (c) Apply spatial reasoning rules to f_i and generate the basic spatial relation set R_i for object pair $(O_1\ O_2)$.

(d) If $R_q \neq R_i$, then GoTo 5.

4. *Result* $\leftarrow$ *Result* $\cup \{i\}$

5. If $i = N$
   then return *Result*
   else $i \leftarrow i + 1$; GoTo 3.

6. Implementation of a Prototype System

Based on the above theories, a prototype system was written in C and implemented on an IBM compatible PC. The major functions of this system are briefly described as follows.

1. *Drawing a picture:*
   The user can draw a picture inside a 640 × 480 picture frame on the screen using a mouse.

2. *Creating rectangles to approximate objects:*
   Once a picture has been drawn on the screen, the next step is to find a set of rectangles to approximate each object. We can move the mouse and position the pointer inside a target object. By clicking the left (right) button, we can increase the number of horizontal (vertical) grid lines. The number of horizontal or vertical lines can be decreased by the same method in a different mode. The merging process is invoked by hitting the space bar on the keyboard. The result from the merging process is a set of approximation rectangles for the target object. The rectangles are highlighted in different colors. The user needs to make sure that the category spatial relationships between any two objects represented by approximation rectangles must be the same as those category spatial relationships among real objects.

3. *Storing a picture in the database:*
   Once all objects are approximated by rectangles, the user can ask the system to generate a RCOS-string for the picture containing highlighted rectangles by hitting the "enter" key. The graphic data and the RCOS-string of the picture are stored into the database and the iconic index area, respectively. The graphic data of a picture and its corresponding RCOS-string are correlated by a unique sequence number.

4. *Deleting pictures from the database:*
   We can delete the graphic data of a picture from the database by entering the sequence number of that picture. Consequently, the corresponding RCOS-string will be removed from the iconic index area too.

5. *Browsing part of a database:*
   By entering the sequence number of a picture or a list of sequence numbers of desired pictures, the system will find the corresponding RCOS-string(s). Then, sketch pictures are reconstructed from RCOS-strings and displayed on the screen one at a time.

6. *Retrieving similar pictures from the database:*
   The user can also draw a query picture on the screen, create grid lines to approximate objects, and ask the system to generate a RCOS-string by the same method as described previously. However, the user must specify that he is in a "query" mode in order to invoke the similarity retrieval algorithm. The result is a list of sequence numbers of matched database pictures. Such sequence numbers are subsequently used to retrieve the corresponding RCOS-strings from which sketch pictures can be reconstructed for browsing. The original pictures may also be retrieved by using these sequence numbers and displayed to the user for further investigation.

7. Conclusions

In recent years, much attention has been paid to design of pictorial database systems. Pictures of arbitrary complexity must be stored and retrieved in a very efficient way. Other important functions of such systems include spatial reasoning, similarity retrieval, visualization/browsing and query processing. The technique of picture abstraction and iconic indexing is the key to fully support these functions [15].

Previous approaches to picture abstraction and iconic indexing use a point or an MBR to represent an object. Consequently, pictures reconstructed from such techniques will have remarkable distortion and the user may not be satisfied with this kind of results for visualization or database browsing. Furthermore, the spatial relationships among objects are very likely to be misjudged or misinterpreted based on MBR approximation. As a result, spatial reasoning and similarity retrieval by spatial relationships based on MBR become inaccurate.

In this paper, we present a "split-and-merge" method which allows us to generate a set of rectangles to approximate the shape of an object. Then, we adopt RCOS-string as the structure of iconic indexes for pictures with objects approximated by rectangles. The number of approximation rectangles of each object can be controlled as small as possible so that the length of a RCOS-string will be short enough to facilitate processing while still maintaining accurate spatial relationships among objects embedded in the original picture. We call this technique as "iconic indexing with shape approximation." We also propose a set of spatial reasoning rules and a similarity retrieval algorithm based on this concept. Finally, we describe a prototype system which was written in C and implemented on IBM compatible PC. The results from this prototype system show that more accuracy can be achieved in picture reconstruction, spatial reasoning and similarity retrieval by our approach.

References

[1] S.K. Chang, Q.Y. Shi, and C.W. Yan, Iconic indexing by 2-D strings, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-9** ,413-428 (May 1987).

[2] S.K. Chang, C.W. Yan, Donald C. Dimitroff, and Timothy Arndt, An intelligent image database

system, *IEEE Trans. on Software Engineering* **14**, 681-688 (May 1988).

[3] E. Jungert, Extended symbolic projections as a knowledge structure for spatial reasoning, *4th BPRA Conf. on Pattern Recognition*, Springer, Cambridge, 343-351 (March 1988).

[4] S.K. Chang, E. Jungert, and Y. Li, Representation and retrieval of symbolic pictures using generalized 2D strings, *Technical Report*, University of Pittsburg (1988).

[5] E. Jungert and S. K. Chang, An algebra for symbolic image manipulation and transformation, in *Visual Database Systems* (edited by T. L. Kunii). Elsevier Science Publishers B. V. (North-Holland), IFIP (1989).

[6] S.Y. Lee and F.J. Hsu, 2D C-string: a new spatial knowledge representation for image database system, *Pattern Recognition* **23**, 1077-1087 (1990).

[7] Chin-Chen Chang and Chin-Feng Lee, Relative coordinates oriented symbolic string for spatial relationship retrieval, *Pattern Recognition* **28**, 563-570 (1995).

[8] S. Y. Lee, M. C. Yang and J. W. Chen, 2D B-string: a spatial knowledge representation for image database systems, *Proc. ICSC92 Int. Computer Sci. Conf.*, 609-615 (1992).

[9] S. K. Chang, *Principles of Pictorial Information Systems Design.* Prentice Hall, Englewood Cliffs, NJ (1989).

[10] S. Y. Lee, M. K. Shan and W. P. Yang, Similarity retrieval of iconic image database, *Pattern Recognition* **22**, 675-682 (1989).

[11] Chin-Chen Chang and S. Y. Lee, Retrieval of similar pictures on pictorial databases, *Pattern Recognition* **24**, 675-680 (1991).

[12] S. Y. Lee and F. J. Hsu, Spatial reasoning and similarity retrieval of images using 2D C-string knowledge representation, *Pattern Recognition* **25**, 305-318 (1992).

[13] P.W. Huang and Y.R. Jean, Using 2D C^+-string as spatial knowledge representation for image database systems, *Pattern Recognition* **27**, 1249-1257 (1994).

[14] P.W. Huang and Y.R. Jean, Reasoning about pictures and similarity retrieval for image information systems based on SK-set knowledge representation, *Pattern Recognition* **28**, 1915-1925 (1995).

[15] S. K. Chang and S. H. Liu, Picture indexing and abstraction techniques for pictorial databases, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-6**, 475-484 (July 1984).