RELATIVE COORDINATES ORIENTED SYMBOLIC STRING
FOR SPATIAL RELATIONSHIP RETRIEVAL

CHIN-CHEN CHANG* and CHIN-FENG LEE
Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan 62107, R.O.C

(Received 4 November 1993; in revised form 10 August 1994; received for publication 1 September 1994)

Abstract—Issues involving efficient data storage and retrieval have become increasingly important in the design of image database systems. The two dimensional B-string, a kind of symbolic picture representation, has been utilized for both characterizing the spatial knowledge embedded in images and for describing images with either partially or completely overlapping objects, without the need to partition objects. A new more efficient data structure, called the Relative Coordinates Oriented Symbolic String (the RCOS String), is proposed in this paper. Two decision trees are constructed so as to obtain the exact spatial relationship between any two objects. The feasibility of the proposed method is confirmed by performing some empirical analyses.

Image databases Iconic indexing 2D strings Spatial relationship 2D B-string

1. INTRODUCTION

Information management and retrieval systems have been a major and interesting part of computer science for a long time. Now, with the increasing requirements of new applications for large amounts of pictorial information (e.g. in office information systems, CAD/CAM systems, geographical information systems, library systems and multimedia database systems), traditional database systems have been extended to image databases which contain large volumes of complex pictures. The main idea of an image database lies in its representation of images and their associated properties by using both physical pictures and logical pictures, respectively. The physical picture can be directly obtained by scanning the real image. The logical picture can be seen as the abstract model of the corresponding physical picture, and it consists of the logical properties of the images. These logical properties can be extracted through image processing techniques. By a table look-up of image features and secondary information which describe the logical pictures, the corresponding physical pictures are retrieved and displayed. However, such database systems are apparently inefficient and ineffective. Therefore, Packed R tree(2) and Intelligent Image Database Systems (IIDS)(3) have proposed, providing a high-level object-oriented search rather than a search based upon low-level image primitives of objects. Chang et al.(5) not only proposed the IIDS to support spatial reasoning, flexible image information retrieval, visualization and traditional image operators, but also introduced a new data structure, called the 2D string, to represent a symbolic picture.

In 1989, Chang et al.(5) introduced the generalized 2D string (2D G-string), a kind of mechanism using a cutting technique which is performed at the extreme points of each object, to segment those objects in the image. In 1990, Lee et al.(6) proposed a set of spatial operators and a new spatial knowledge representation, the 2D C-string, which preserves all spatial relations among objects with more efficient segmentation for image database systems. 2D C-strings have proven to be more efficient than 2D G-strings in both storage space as well as spatial reasoning complexity.

For the various 2D string representations mentioned so far, objects may be partitioned into subparts so as to represent the spatial relations among the objects—especially for the case of overlapping objects. Once an object is partitioned into subparts and stored in the data structure, the several subparts must be treated together as a whole for integrally performing spatial inferences about the objects. However, the storage space overhead and processing time would be excessive for a large number of subparts. Therefore, in 1992, Lee et al.(7) proposed yet another new spatial knowledge representation called a 2D B-string for improving the performance of spatial inferences. By using the ranks of symbols, 2D B-strings can preserve all of the essential spatial information while, at the same time, supporting the indexing of those images without the need to partition objects into subparts.

A new data structure is proposed in this paper, termed the Relative Coordinates Oriented Symbolic String (the RCOS String) from the boundary concept.
to characterize the spatial knowledge embedded in images by 2D B-strings. Each object in this structure can be represented by its beginning boundary and end boundary so as to be able to represent the spatial relationship between any two objects in an image. From the proposed data structure, the boundaries of each object can be extracted in linear time. Additionally, one exact type of the 169 possible spatial relationships can be obtained after tracing through two decision trees, each corresponding to a one-dimensional space.

The rest of this paper is organized as follows. The 2D string and the 2D B-string approaches are reviewed in Section 2. Our new data structure for spatial-relationship retrieval is presented in Section 3. Some empirical analyses are then presented in Section 4. Concluding remarks are finally provided in Section 5.

2. A REVIEW OF 2D STRINGS AND 2D B-STRINGS

This section briefly reviews 2D strings and 2D B-strings. For more details, refer to References (4) and (5).

Image processing and pattern recognition techniques are initially required to recognize objects along with their corresponding symbolic names in an original image. A non-zero sized object in a symbolic picture is enclosed by a Minimum Bounding Rectangle (MBR), with boundaries parallel to the x- and y-axis. The symbolic picture in Fig. 1 has four symbols and each symbol corresponds to a pictorial object. All possible combinations are presented as follows, where the empty set \{\} denotes a null object.

\[
\begin{align*}
&f(1, 1) = \{A\}, f(2, 1) = \{B\}, f(3, 1) = \{\}\, , \\
&f(1, 2) = \{\} , f(2, 2) = \{\} , f(3, 2) = \{D\} , \\
&f(1, 3) = \{\} , f(2, 3) = \{C\} , f(3, 3) = \{\} .
\end{align*}
\]

With such a representation, the spatial relationships among these four objects are clear. For instance, object C is to the north of object B and object A is to the south-west of object D.

Let \(S = \{O_1, O_2, \ldots, O_N\}\) be a set of symbols, where \(O_i\) represents a pictorial object, for \(i = 1, 2, \ldots, N\). Let \(R = \{=, <, >\}\) denote a set of relations, where each element in \(R\) represents the spatial relationship between two objects. Here "=" represents the spatial relation "at the same x or y location as", "<" represents the spatial relation "to the west of or the south of" and ">" represents the relation "in the same set as".

A 2D string over \(S \cup R\) is represented as \((O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(r_1, r_2, \ldots, r_N^{-1}\) are relations in set \(R\) and \(p\) is a permutation function from \(\{1, 2, \ldots, N\}\) to \(\{1, 2, \ldots, N\}\). Each symbolic picture of a pictorial match query is capable of being represented by a 2D string. Figure 2 is used as an example in the following.

The set of symbols \(S = \{A, B, C, D, E\}\). The 2D string \((A < B : C = D < E, B: C = A < E < D) = (O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(O_1, O_2, O_3, O_4, O_5\) are \(A, B, C, D, E\), respectively, \(p(1) = p(2) = p(3) = p(4) = p(5) = \Lambda\), \(B, C, E, D\), respectively.

As observed from the example in Fig. 2, the two 1D strings, "A < B : C = D < E" and "B = C = A < E < D", are the symbolic projection of objects A, B, C, D and E along the x-axis and the y-axis, respectively.

For the case of non-overlapping, non-zero sized objects, the projection in the x- or y-axis is non-overlapping ("<") or at the same position ("=" or "="). However, for the case of completely-overlapping or partially-overlapping non-zero sized objects, an object may be segmented into many subparts by a cutting mechanism, as introduced in 2D G-strings(5) and 2D C-strings.(6) The partition process is quite time-consuming, and it also creates a large storage overhead for the spatial reasoning on spatial queries. The 2D B-string approach, on the other hand, does not require that the objects be partitioned. The spatial relationships are derived by using the ranks of symbols in the 2D B-string representation as follows.(7,8)

Let \(S'\) be a set of object symbols, in which each symbol denotes the beginning boundary or end boundary of a physical pictorial object. The projection of each object along the x- or y-axis is described by its two boundaries, i.e. the beginning and the end boundaries. The special symbol "=" (not in the set \(S'\)) specifies that the projections of two objects either have the same boundary or is at the same location.

A 2D B-string over \(S' \cup \{\} = \{A, B, C, D, E\}\) is represented as \((O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(r_1, r_2, \ldots, r_N^{-1}\) are relations in set \(R\) and \(p\) is a permutation function from \(\{1, 2, \ldots, N\}\) to \(\{1, 2, \ldots, N\}\). Each symbolic picture of a pictorial match query is capable of being represented by a 2D B-string. Figure 2 is used as an example in the following.

The set of symbols \(S = \{A, B, C, D, E\}\). The 2D string \((A < B : C = D < E, B: C = A < E < D) = (O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(O_1, O_2, O_3, O_4, O_5\) are \(A, B, C, D, E\), respectively, \(p(1) = p(2) = p(3) = p(4) = p(5) = \Lambda\), \(B, C, E, D\), respectively.

As observed from the example in Fig. 2, the two 1D strings, "A < B : C = D < E" and "B = C = A < E < D", are the symbolic projection of objects A, B, C, D and E along the x-axis and the y-axis, respectively.

For the case of non-overlapping, non-zero sized objects, the projection in the x- or y-axis is non-overlapping ("<") or at the same position ("=" or ">"). However, for the case of completely-overlapping or partially-overlapping non-zero sized objects, an object may be segmented into many subparts by a cutting mechanism, as introduced in 2D G-strings(5) and 2D C-strings.(6) The partition process is quite time-consuming, and it also creates a large storage overhead for the spatial reasoning on spatial queries. The 2D B-string approach, on the other hand, does not require that the objects be partitioned. The spatial relationships are derived by using the ranks of symbols in the 2D B-string representation as follows.(7,8)

Let \(S'\) be a set of object symbols, in which each symbol denotes the beginning boundary or end boundary of a physical pictorial object. The projection of each object along the x- or y-axis is described by its two boundaries, i.e. the beginning and the end boundaries. The special symbol "=" (not in the set \(S'\)) specifies that the projections of two objects either have the same boundary or is at the same location.

A 2D B-string over \(S' \cup \{\} = \{A, B, C, D, E\}\) is represented as \((O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(r_1, r_2, \ldots, r_N^{-1}\) are relations in set \(R\) and \(p\) is a permutation function from \(\{1, 2, \ldots, N\}\) to \(\{1, 2, \ldots, N\}\). Each symbolic picture of a pictorial match query is capable of being represented by a 2D B-string. Figure 2 is used as an example in the following.

The set of symbols \(S = \{A, B, C, D, E\}\). The 2D string \((A < B : C = D < E, B: C = A < E < D) = (O_1r_1O_2r_2 \ldots r_N^{-1}O_N)\), where \(O_1, O_2, O_3, O_4, O_5\) are \(A, B, C, D, E\), respectively, \(p(1) = p(2) = p(3) = p(4) = p(5) = \Lambda\), \(B, C, E, D\), respectively.

As observed from the example in Fig. 2, the two 1D strings, "A < B : C = D < E" and "B = C = A < E < D", are the symbolic projection of objects A, B, C, D and E along the x-axis and the y-axis, respectively.

For the case of non-overlapping, non-zero sized objects, the projection in the x- or y-axis is non-overlapping ("<") or at the same position ("=" or ">"). However, for the case of completely-overlapping or partially-overlapping non-zero sized objects, an object may be segmented into many subparts by a cutting mechanism, as introduced in 2D G-strings(5) and 2D C-strings.(6) The partition process is quite time-consuming, and it also creates a large storage overhead for the spatial reasoning on spatial queries. The 2D B-string approach, on the other hand, does not require that the objects be partitioned. The spatial relationships are derived by using the ranks of symbols in the 2D B-string representation as follows.(7,8)

Let \(S'\) be a set of object symbols, in which each symbol denotes the beginning boundary or end boundary of a physical pictorial object. The projection of each object along the x- or y-axis is described by its two boundaries, i.e. the beginning and the end boundaries. The special symbol "=" (not in the set \(S'\)) specifies that the projections of two objects either have the same boundary or is at the same location.
Definition 1. The rank of each object symbol

The rank of each object symbol in a string \(U_x\) (or string \(U_y\)) is defined as the position of this object symbol minus the number of symbols "=" proceeding this objects symbol in the string \(U_x\) (or the string \(U_y\)). The position of the first object in a 2D B-string is set to one and the symbol "=" is not counted in the calculation of the positions of object symbols. The ranks of each object can be represented as \((i, j, k, l)\), where \((i, j)\) and \((k, l)\) are the ranks of the beginning and the end boundaries along the x- and y-axis directions, respectively.

Take Fig. 3(a) as an example. The 2D B-string representation around the positions of those objects A, B and C in the strings \(U_x\) and \(U_y\) are "ABABCC" and "AABBCC", respectively. The beginning rank (denoted by \(rb\)) and the end rank (denoted by \(re\)) for each object are listed as follows:

\[\begin{align*}
U_x: \\
rb(A) &= 1, \quad re(A) = 3, \\
rb(B) &= 2, \quad re(B) = 4, \\
rb(C) &= 4, \quad re(C) = 5,
\end{align*}\]

\[\begin{align*}
U_y: \\
rb(A) &= 1, \quad re(A) = 2, \\
rb(B) &= 3, \quad re(B) = 4, \\
rb(C) &= 4, \quad re(C) = 5,
\end{align*}\]

This yields:

\[\begin{align*}
\text{rank}(A) &= (1, 3, 1, 2), \\
\text{rank}(B) &= (2, 4, 3, 4), \\
\text{rank}(C) &= (4, 5, 4, 5).
\end{align*}\]

The 2D B-string representation can obviously preserve both the spatial knowledge of the relative topological sequencing and the ordering of the objects embedded in the original image. Figure 3(b) illustrates the reconstruction of a symbolic picture by using the ranks of the 2D B-string representation (ABAB = CC, AABB = CC).

Allen\(^9\) has presented thirteen varieties of spatial relationships between two objects in one dimension, as shown in Fig. 4.

If x- and y-direction are considered independently, a total of 169 spatial relationships arise between two objects in two dimensions.\(^{7-9}\)

Some techniques for subpicture matching are necessary for retrieving pictures from the image database. By using a 2D B-string representation for a symbolic picture, the problem of subpicture matching is converted into the problem of subsequence matching. Hence, the following definition, provided by Lee et al., functions as the similarity criterion of a query retrieval.

Definition 2. Three types of similarity measures\(^{7-8}\).

Let \(A C_{AB} B\) denote a category type of five classes (disjoint, meet, contain, inside and partial overlap) between A and B, let \(A O_{AB} B\) denote an orthogonal relation between A and B, and let \(A R_{AB} B\) denote one of the 169 varieties of spatial relationships in a two dimensional space.

Picture \(f'\) is a type-i subpicture of \(f\) if:

1. all objects in \(f'\) are also in \(f\);
2. for any two objects, \(A\) and \(B\):
   \(A C_{AB} B, A O_{AB} B\) and \(A R_{AB} B\) are in \(f\) and
   \(A C'_{AB} B, A O'_{AB} B\) and \(A R'_{AB} B\) are in \(f'\), then
   (type-0) \(C_{AB} = C'_{AB}\)
   (type-1) \(C_{AB} = C_{AB}\) and \(O_{AB} = O'_{AB}\)
   (type-2) \(C_{AB} = C_{AB}\) and \(O_{AB} = O_{AB}\) and \(R_{AB} = R_{AB}\)

Algorithm type-i has been proposed by Lee et al. for checking each type-i (\(i = 0, 1\) and 2) similarity of object pair in \(f\) and \(f'\). A more detailed discussion is provided in Reference (7).

3. A RELATIVE COORDINATES SYMBOLIC STRING

In this section, a new data structure called the Relative Coordinates Oriented Symbolic (RCOS) string...
representation is proposed to maintain the advantages of 2D B-strings, in which a spatial knowledge is preserved without any object being segmented. Moreover, the RCOS strings also can check whether a query precisely matches the desired image with one of the 169 varieties of spatial relations between each two objects.

Let \( S = \{O_1, O_2, \ldots, O_N\} \) be the collection of object symbols in a symbolic picture. Since each symbolic object \( O_i \) is enclosed by an MBR with boundaries parallel to the horizontal \((x-)\) and vertical \((y-)\) axis, \((x_{i1}, y_{i1})\) and \((x_{i2}, y_{i2})\) that can express the position of the object symbol \( O_i \). That is, two projections which represent the beginning and the end boundaries are available for each object along each axis. Let \( x_{i1} \) and \( x_{i2} \) denote the beginning rank and the end rank along the \( x \)-axis, respectively, and let \( y_{i1} \) and \( y_{i2} \) denote the beginning rank and the end rank along the \( y \)-axis, respectively, where the "rank" is defined in Definition 1 (as described in Section 2). The \((x_{i1}, y_{i1})\) and \((x_{i2}, y_{i2})\) are used as the coordinates corresponding to the object symbol \( O_i \), where \( (x_{i1}, y_{i1}) \) denotes the lower-left coordinate and \( (x_{i2}, y_{i2}) \) denotes the upper-right coordinate of the symbolic object \( O_i \) enclosed by an MBR.

For example, in Fig. 5, \( S = \{A, B, C, D\} \). The lower-left and upper-right coordinates corresponding for object A are \((1, 1)\) and \((3, 3)\), for object B are \((2, 2)\) and \((4, 4)\), for object C are \((4, 4)\) and \((5, 5)\) and for object D are \((4, 4)\) and \((6, 6)\).

A new data structure called the RCOS string is then proposed in the following. If there are \( N \) symbolic objects in a symbolic picture \( f \), let \( S = \{O_1, O_2, \ldots, O_N\} \) denote the set of these \( N \) symbolic objects. Since each symbolic object \( O_i \) has two pairs of coordinates \((x_{i1}, y_{i1})\) and \((x_{i2}, y_{i2})\) to denote its position, the symbolic picture \( f \) can be represented by \([O_1, O_2, \ldots, O_N, (x_{i1}, y_{i1}), (x_{i2}, y_{i2}), (x_{j1}, y_{j1}), (x_{j2}, y_{j2}), \ldots, (x_{N1}, y_{N1}), (x_{N2}, y_{N2})]\), an ordered list of length \( 5*N \). Each term is delimited by the comma "," in the RCOS string. The first term consists of the \( N \) symbolic objects appearing in the symbolic picture \( f \). Two entries, i.e., \((x_{i1}, y_{i1})\) and \((x_{i2}, y_{i2})\), are used for the \((i+1)th\) term, which represents a pair of relative coordinates corresponding to the symbolic object \( O_i \). The case shown in Fig. 5 is an example, in which the new data structure is expressed as an ordered list of \([ABCD, (1, 1)(3, 3), (2, 2)(4, 4), (4, 4)(5, 5), (4, 4)(6, 6)]\), where A, B, C and D are four symbolic objects in picture \( f \) with \((1, 1)\) and \((3, 3)\) denoting a pair of relative coordinates corresponding to A, \((2, 2)\) and \((4, 4)\) denoting a pair of relative coordinates corresponding to B, \((4, 4)\) and \((5, 5)\) denoting a pair of relative coordinates corresponding to C and \((4, 4)\) and \((6, 6)\) denoting a pair of relative coordinates corresponding to D, respectively.

Before introducing the applications of RCOS strings, a decision tree (called an \( x \)-decision tree) in Fig. 6 is presented to specify thirteen varieties of spatial relationships between two objects in a 1D space as shown in Fig. 4. Figure 6 depicts all of the possibilities covering the complete thirteen relations in the \( x \) direction between objects \( O_i \) and \( O_j \) with the relative coordinates \((x_{i1}, y_{i1})\) and \((x_{i2}, y_{i2})\) and the relative coordinates \((x_{j1}, y_{j1})\) and \((x_{j2}, y_{j2})\), respectively, where \( i \neq j \) and \( i, j \in \{1, 2, \ldots, N\} \). Each path requires at most three comparisons in the \( x \)-decision tree. Moreover, each leaf node would denote one type among the thirteen varieties of spatial relationships if the \( x \)-coordinates of two objects are utilized in constructing the \( x \)-decision tree. For more convenient description, each leaf node from left to right in the \( x \)-decision tree is denoted as \( n_i \), for \( i = 1 \) to 13. Therefore, of all the thirteen spatial relation types between any two objects as shown in Fig. 4, each
Relative coordinates oriented symbolic string

Fig. 7. The y-decision tree constructed according to the y-coordinates of objects O_i and O_j.

Fig. 8. The 169 types of spatial relations in two dimensions.
is permitted to retrieve the picture in Fig. 5. The symbol
Q3 is found to be the only candidate query which
as follows:

\[ \text{Q3} = \text{[ABC, (1, 3), (2, 2)(3, 3)]} \]

Then Q3 is found to be the only candidate query which
is permitted to retrieve the picture in Fig. 5. The symbol

4. SOME EMPIRICAL ANALYSES

<table>
<thead>
<tr>
<th>Number of symbols</th>
<th>2D B-string</th>
<th>RCOS string</th>
<th>R/B</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8.0</td>
<td>10.0</td>
<td>1.25</td>
</tr>
<tr>
<td>4</td>
<td>16.0</td>
<td>20.0</td>
<td>1.25</td>
</tr>
<tr>
<td>8</td>
<td>32.0</td>
<td>40.0</td>
<td>1.25</td>
</tr>
<tr>
<td>16</td>
<td>64.0</td>
<td>80.0</td>
<td>1.25</td>
</tr>
<tr>
<td>32</td>
<td>128.0</td>
<td>160.0</td>
<td>1.41</td>
</tr>
<tr>
<td>64</td>
<td>256.0</td>
<td>320.0</td>
<td>1.25</td>
</tr>
<tr>
<td>128</td>
<td>512.0</td>
<td>640.0</td>
<td>1.25</td>
</tr>
</tbody>
</table>

The results of the 2D B-string come from the simulation result depicted by the Table 1 in Reference (7).
Table 2. The comparison results among the 2D G-string, the 2D C-string and the RCOS string with R denoting the RCOS string, G denoting the 2D G-string and C denoting the 2D C-string, respectively.

<table>
<thead>
<tr>
<th>Number of objects (N)</th>
<th>Number of symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D G-string</td>
</tr>
<tr>
<td>2</td>
<td>10.0</td>
</tr>
<tr>
<td>4</td>
<td>43.6</td>
</tr>
<tr>
<td>8</td>
<td>173.2</td>
</tr>
<tr>
<td>16</td>
<td>672.0</td>
</tr>
<tr>
<td>32</td>
<td>2673.2</td>
</tr>
<tr>
<td>64</td>
<td>9626.0</td>
</tr>
<tr>
<td>128</td>
<td>32419.2</td>
</tr>
</tbody>
</table>

The results of the 2D G-string and the 2D C-string come from the simulation result depicted by the Table 1 in Reference (7).

Table 3. The number of operations between 2D B-strings and RCOS string while retrieving one among the 169 varieties of possible spatial relationship types in a two dimensional space.

<table>
<thead>
<tr>
<th>Number of operations</th>
<th>2D B-string</th>
<th>RCOS string</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of comparisons</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>No. of subtractions</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>No. of multiplications</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

According to the simulation results shown in Table 2, our proposed data structure is better than 2D G-strings and 2D C-strings, and it will be better than 2D B-strings when the number of symbols is large. This enhanced performance occurs as a result of the ratios quickly decreasing as the number of objects increases. Moreover, the RCOS string representation is well-formatted with a fixed length and can be processed in linear time. Besides, Table 3 depicts the number of operations for similarity retrieval by using the 2D B-string and the RCOS string. The latter is observed to greatly reduce the number of operations since at most six comparison operations are needed for RCOS strings while sixteen subtractions and eight multiplications are required in the 2D B-string algorithm proposed in Reference (7).

5. CONCLUSIONS

Extracting the information on the relative positions of objects and storing it properly allows queries concerning the relative positions of objects in an image to be processed more efficiently. Additionally, queries are usually processed by consulting such information instead of the whole image itself. Symbolic pictures have been proposed for describing the relations among objects in image databases. Lee et al.\(^{(7)}\) proposed the 2D B-string representation to preserve all spatial relations among objects, without segmentation, and the spatial relationships can be easily derived by using the ranks of 2D B-strings. Lee et al.\(^{(7,8)}\) also presented some measure criteria of three types of similarity retrieval. However, the most precise spatial relational representation among objects is the type-2, which denotes the similarity retrieval in 169 varieties of possible spatial relationships in a 2D space. A new data structure, i.e. the Relative Coordinates Oriented Symbolic String (RCOS String), has therefore been proposed in this present work which is another alternative for representing the spatial relationship between two symbolic objects in an image. Based on the projection of an object by its boundaries as well as 2D B-strings, no cutting mechanism is needed to solve objects with overlapping. The coordinates of an object symbol are extracted from the RCOS string representation and the x-coordinates of two objects O\(_i\) and O\(_j\) are applied toward the x-decision tree constructed by using the x-coordinate first and then the y-coordinates of objects O\(_i\) and O\(_j\) are applied toward the y-decision tree constructed by using the y-coordinate. After tracing through the two decision trees, the precise type of the 169 spatial relationship kinds in two dimensional space is obtained immediately without type-0 and then type-1 similarity checking processing as introduced in 2D B-strings. The efficiency between the 2D B-string representation and the RCOS string representation has also been confirmed by the empirical analyses undertaken here. From the results of the average number of symbols to store the 2D B-strings and the RCOS strings, the former performs better than the latter. However, the ratio of the average number of symbols in the RCOS string to that in the 2D B-string obviously decreases quickly when the number of objects increases. Moreover, the RCOS string is well-formatted with a fixed length, and it can be processed in linear time. For instance, such a data structure can be used as a kind of filter in either searching for the symbolic string terms or else to facilitate the extraction of the relative coordinates corresponding to a symbolic object.

A comparison of the operations between the 2D B-string approach and ours indicates that the RCOS string approach greatly reduces the number of operations, since at most six comparison operations are needed, while sixteen subtractions and eight multiplications are needed for the 2D B-string algorithm proposed in Reference (7).

Finally, the two decision trees introduced here may notably occupy some storage space; however, saving
the total computation time is more important for a frequently accessed image database. Thus, the storage overhead for the RCOS string preprocessing is tolerable. 

Acknowledgment—The authors would like to express their gratitude for the referees' excellent suggestions.

REFERENCES


About the Author—CHIN-CHEN CHANG was born in Taichung, Taiwan, Republic of China, on November 12, 1954. He received his B.S. degree in Applied Mathematics in 1977 and his M.S. degree in Computer and Decision Sciences in 1979 from National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D. degree in Computer Engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. During the academic years 1980–83, he was on the faculty at the Department of Computer Engineering at National Chiao Tung University. From 1983 to 1989, he was on the faculty at the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head and professor of the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiaiyi, Taiwan, R.O.C. Since August 1992, he has been the Dean of the College of Engineering at National Chung Cheng University. In addition, he has served as consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, coding theory and data structures.


Dr Chang is a senior member of the IEEE, and a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, the International Association for Cryptologic Research, the Computer Society of the Republic of China, and the Phi Tau Phi Society of the Republic of China and he is also the Chairman of the Executive Committee in the Information Security Society of the Republic of China.

About the Author—CHIN-FENG LEE is currently a student in the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiaiyi, Taiwan, R.O.C. She received her B.S. degree in 1992 and M.S. degree in 1994 in Information Engineering and Computer Science at National Taiwan University and at National Chung Cheng University, respectively. Her main research interests include database and data engineering design.