Efficient Access Methods for Image Databases [1]

Ye-In Chang and Bi-Yen Yang

Dept. of Applied Mathematics
National Sun Yat-Sen University
Kaohsiung, Taiwan
R.O.C.
{E-mail: changyi@math.nsysu.edu.tw}
{Tel: 887-7-5252000 (ext. 3819)}
{Fax: 886-7-5253809}

Abstract

In the previous approaches to retrieve a symbolic picture from an image database which contains multiple symbolic pictures, they only consider 9 spatial relationships that are represented in 2D strings or 9DLT matrices. In this paper, we propose two efficient access methods for image databases, which can handle 169 spatial relationships. In the first proposed method, each record signature is represented by 26 bits and 26 bit strings of size N, where $N > 0$. In the second proposed method, each record signature is represented by 26 bits and 26 products of prime numbers. Given the same image databases, the same query picture and the same hash functions, the ratio of the false match in our strategy can be smaller than that of the previous approaches, where a false match is that that a record signature matches a query signature but the corresponding record does not satisfy the query.

(*Keywords*: 2D string, 2D C-string, image databases, pictorial query, similarity retrieval, spatial reasoning, symbolic databases)

[1] This research was supported in part by the National Science Council of Republic of China under Grant No. NSC-87-2213-E-110-014.

1 Introduction

The perception of spatial relationships among objects in a picture is one of important selection criteria to discriminate and retrieve images in an image database system. When there are a large number of images in the image database and each image contains many objects, the processing time for image retrievals is tremendous. Therefore, to handle large amounts of image databases, several access methods [1, 2, 4] have been proposed for image databases by using the concept of *superimposed coding* [3] and two-level signature files [6].

Lee et al. [4] proposed a method for retrieval by subpictures that are represented in 2D strings. In Lee et al.'s strategy for image retrieval [4], they consider only three spatial operators along the x-axis or the y-axis; it is too rough. To reduce the ratio of a false match and to reduce the number of needed record signatures and block signatures, where a false match is that that a record signature matches a query signature but the corresponding record does not satisfy the query, Chang et al. proposed a strategy with pictures represented in 9DLT matrices, in which new data structures of record signature and block signature are developed. Note that the number of record signatures for spatial strings used in Lee et al.'s strategy [4] is 9 times of that used in Chang et al.'s strategy [1] in the worst case, since for each record signature for a pattern $ABij$ in Lee et al.'s strategy, $1 \leq i, j \leq 3$, only one record signature for a pattern AB is used in Chang et al.'s strategy. In Chang et al.'s strategy, each record signature is represented by 9 bits and 9 bit strings of size N, where $N > 0$.

In the previous approaches to retrieve a symbolic picture from an image database which contains multiple symbolic pictures, they only consider 9 spatial relationships that are represented in 2D strings or 9DLT matrices. In this paper, we propose two efficient access methods for image databases, which can handle 169 spatial relationships. In the first proposed method, each record signature is represented by 26 bits and 26 bit strings of size N, where $N > 0$, instead of 169 bits and 169 bit strings of size N if we directly apply Chang et al.'s strategy [1] to record 169 spatial relationships. In the second proposed method, each record signature is represented by 26 bits and 26 products of prime numbers. Given the same image databases, the same query picture and the same hash functions, the ratio of false match in our strategy can be smaller than that of the previous approaches.

2 A Bit-String-Signature-Based Access Method

2.1 A Reduced Spatial Matrix

The 2D C-string is proposed by Lee and Hsu [5]. Table 1 shows the formal definition of the set of spatial operators defined in the 2D C-string representation, where the notation "begin(A)" denotes the value of begin-bound of object A and "end(A)" denotes the value of end-bound of object A. According to the begin-bound and end-bound of the picture objects, spatial relationships between two enclosing rectangles can be categorized into 13 types ignoring their length along the x-(or y-) axis. Therefore, There are 169 types of spatial relationships between two rectangles in 2D space, as shown in Figure 1.

Suppose a picture f contains m objects and let $V = \{v_1, v_2, ..., v_m\}$. Let R be the set of 13 spatial operators $\{<, <^*, |, |^*, [, [^*,],]^*, \%, \%^*, /, /^*, =\}$ defined in 2D C-string [5], where operator* means the inverse operator of the related operator. A $m \times m$ *spatial matrix* S of picture f is defined as follows:

Notation	Condition	Meaning
A < B	end(A) < begin (B)	A disjoins B
A = B	begin(A) = begin(B) end(A) = end(B)	A is the same as B
A | B	end(A) = begin(B)	A is edge to edge with B
A % B	begin(A) < begin(B) end(A) > end(B)	A contains B and they have not the same bound
A [B	begin(A) = begin(B) end(A) > end(B)	A contains B and they have the same begin bound
A] B	begin(A) < begin(B) end(A) = end(B)	A contains B and they have the same end bound
A / B	begin(A) < begin(B) < end(A) < end(B)	A is partly overlapping with B

Table 1: Definitions of Lee et al.'s spatial operators

Figure 1: The 169 spatial relationship types of two objects

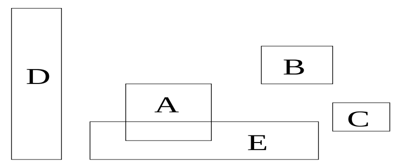

Figure 2: One example of picture f

$$S = \begin{array}{cc} & \begin{array}{ccccc} v_1 & v_2 & \cdots & v_{m-1} & v_m \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ \vdots \\ v_{m-1} \\ v_m \end{array} & \left[\begin{array}{ccccc} 0 & r^y_{1,2} & \cdots & \cdots & r^y_{1,m} \\ r^x_{1,2} & 0 & \ddots & & \vdots \\ \vdots & \ddots & 0 & \ddots & \vdots \\ \vdots & & \ddots & 0 & r^y_{m-1,m} \\ r^x_{1,m} & \cdots & \cdots & r^x_{m-1,m} & 0 \end{array}\right] \end{array}$$

where the lower triangular matrix stores the spatial information along the x-axis, and the upper triangular matrix stores the spatial information along the y-axis. That is, $S[v_i, v_j] = r^x_{j,i}$ if $i > j$; $S[v_i, v_j] = r^y_{i,j}$ if $i < j$; $S[v_i, v_j] = 0$ if $i = j$, $\forall v_i, v_j \in V$, $\forall r^x_{j,i}, r^y_{i,j} \in A$, $1 \leq i, j \leq m$, where $r^x_{j,i}$ is the spatial operator between objects v_i and v_j along the x-axis and $r^y_{i,j}$ is the spatial operator between objects v_i and v_j along the y-axis. Note that in this representation, we always record the relationships between two objects v_i and v_j from the view point of object v_i no matter along the x-axis or the y-axis, where $i < j$. That is why $S[v_i, v_j] = r^x_{ji}$ when $i > j$.

For the picture shown in Figure 2, the corresponding spatial matrix S is shown as follows:

$$S = \begin{array}{cc} & \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} & \left[\begin{array}{ccccc} 0 & | & \% & \%^* & /^* \\ < & 0 & <^* & \%^* & <^* \\ < & | & 0 & \%^* & /^* \\ <^* & <^* & <^* & 0 & [\\ \%^* & /^* & <^* & < & 0 \end{array}\right] \end{array}$$

We let 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13 denote spatial operators $<$, $<^*$, $|$, $|^*$, $[$, $[^*$, $]$, $]^*$, $\%$, $\%^*$, $/$, $/^*$ and $=$, respectively. Then, the corresponding *reduced spatial matrix* (RSM) is as follows:

$$RSM = \begin{array}{cc} & \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} & \left[\begin{array}{ccccc} 0 & 3 & 9 & 10 & 12 \\ 1 & 0 & 2 & 10 & 2 \\ 1 & 3 & 0 & 10 & 12 \\ 2 & 2 & 2 & 0 & 5 \\ 10 & 12 & 2 & 1 & 0 \end{array}\right] \end{array}$$

Moreover, a spatial x-string set T^x is defined as $\{v_j v_i r^x_{ij} | 1 \leq j < i \leq m, r^x_{ji} \in \{1, 2, \cdots, 12, 13\}\}$, and a spatial y-string set T^y is defined as $\{v_i v_j r^y_{ij} | 1 \leq i < j \leq m, r^y_{ij} \in \{1, 2, \cdots, 12, 13\}\}$. Therefore, the corresponding spatial x-string set for the above RS is $T^x = \{$ AB1, AC1, AD2, AE10, BC3, BD2, BE12, CD2, CE2, DE1 $\}$ and the corresponding spatial y-string set for the above RS is $T^y = \{$ AB3, AC9, AD10, AE12, BC2, BD10, BE2, CD10, CE12, DE5 $\}$.

Table 2: The notations and the related definitions used in the proposed access method

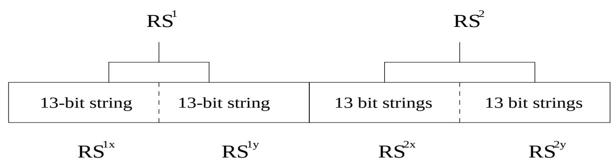

Figure 3: The components of a Record Signature

2.2 Record Signatures and Black Signatures

Based on the 13 spatial operators, we propose a new definition of a Record Signature (RS) and a Block Signature (BS). A RS consists of two parts, RS^1 and RS^2 as shown in Figure 3. RS^1 contains 2 segments RS^{1x} and RS^{1y}, which represent the record signature flags from the view point of x-axis and y-axis, respectively, and each segment is a 13-bit string. These two 13-bit strings are used to indicate the existence or absence of those 13 spatial operators along the x-axis and y-axis, respectively. RS^2 consists of two segments, RS^{2x} and RS^{2y}. Each of these two segments contains 13 bit strings. The i-th bit string (or signature) among those 13 bit strings is used to record the union of the signatures of those pairs of objects which have the same i-th spatial operator. We use $RS^{2x}(i)$ to represent the i-th segment of RS^{2x}. That is, $RS^{2x}(i) = \cup\, \theta_r(XYi)$, X, Y $\in$ V, $i \in \{$ 1, 2, $\cdots$, 12, 13 $\}$, where θ_r is the hash function of the record signature. BS is treated in the same way. The proposed algorithm for efficient data access of image databases is described as follows, where the notations and related definitions used in the algorithm is shown in Table 2.

Algorithm (Record Signature)

(Step 1) According to the reduced spatial matrix, list all the spatial x-string sets T^x and spatial y-string sets T^y.

(Step 2) Design the function θ_r according to the given k_r and b_r, which maps each pair of symbols into a unique bit string.

(Step 3) Set all bits in RS to 0.

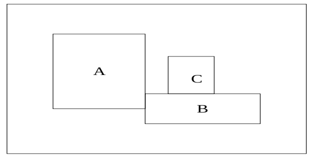

Figure 4: One example

(*Step 4*) *For each spatial x-string ABi in T^x, we let the i-th bit of RS^{1x} be 1, and then perform $RS^{2x}(i) = RS^{2x}(i) \cup \theta_r(AB)$.*

(*Step 5*) *Repeat Step 4 by replacing T^x with T^y.*

(*Step 6*) *Compress RS^2 by removing useless bit strings. If the i-th bit of RS^{1x} (or RS^{1y}) is 0, then remove $RS^{2x}(i)$ (or $RS^{2y}(i)$).*

To reduce the number of comparison with each record signature in the image database, a block signature (BS) is used. The algorithm to find BS is almost the same as RS. The only one difference between them is that we use another function θ_b according to the given k_b and b_b to get the block signatures of object blocks.

To illustrate the algorithm, let's see the following example. For the figure shown in Figure 4, first, we construct the spatial matrix and the reduced spatial matrix (RSM).

$$S = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & /^* & \% \\ | & 0 & | \\ < & \% & 0 \end{array}\right] \end{array} \qquad RSM = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & 12 & 9 \\ 3 & 0 & 3 \\ 1 & 9 & 0 \end{array}\right] \end{array}$$

Applying the algorithm, we can construct the Record Signature of the picture as follows.

(1) Generate the spatial x-string set T^x and spatial y-string set T^y.

T^x = { AB3, AC1, BC9 }.
T^y = { AB12, AC9, BC3 }.

(2) Design the function θ_r (where $b_r = 5$, $k_r = 2$) which maps each pair of symbols to a unique bit string.

spatial string	θ_r
AB	10001
AC	10100
BC	01100

(3) Set all bits in RS to 0.

$RS = RS^1 + RS^2$
= 0000000000000 0000000000000 00000 00000 00000 00000 00000 00000
00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
00000 00000 00000 00000 00000 00000 00000 00000 00000.

(4) If $ABi \in T^x$, we let the i-th bit of RS^{1x} be 1, and then perform $RS^{2x}(i) = RS^{2x}(i) \cup \theta_r(AB)$.

RS^{1x} = 1010000010000.
$RS^{2x}(1) = RS^{2x}(1) \cup \theta_r(AC) = 10100$.
$RS^{2x}(3) = RS^{2x}(3) \cup \theta_r(AB) = 10001$.
$RS^{2x}(9) = RS^{2x}(9) \cup \theta_r(BC) = 01100$.

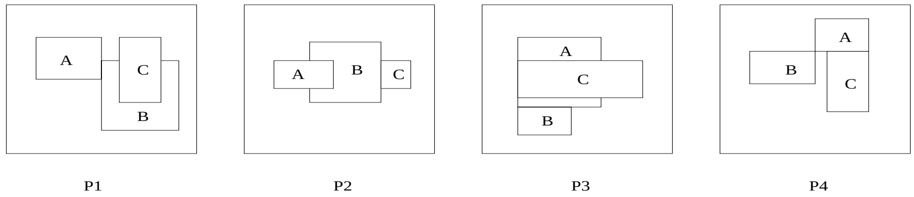

Figure 5: A 4-picture image database

(5) Repeat Step 4 by replacing T^x by T^y. We have
$RS^{2y}(3) = RS^{2y}(3) \cup \theta_r(BC) = 01100.$
$RS^{2y}(9) = RS^{2y}(9) \cup \theta_r(AC) = 10100.$
$RS^{2y}(12) = RS^{2y}(12) \cup \theta_r(AB) = 10001.$

(6) Compress RS^2 by removing useless bit strings. If the i-th bit of RS^{1x} (or RS^{1y}) is 0, then remove $RS^{2x}(i)$ (or $RS^{2y}(i)$).
$RS =$ 1010000010000 001000001001 10100 10001 01100 01100 10100 10001.

2.3 An Example

Suppose there are four pictures in the database P_1, P_2, P_3 and P_4 as shown in Figure 5. Suppose we want to get the pictures with the spatial relationship (AC7, AC4). There are two blocks; B_1 contains P_1 and P_2, and B_2 contains P_3 and P_4, where $N_r = 2$. And the hash functions θ_r (where $b_r = 5$, $k_r = 2$) and θ_b (where $b_b = 7$, $k_b = 1$) are defined as follows.

spatial string	θ_r	θ_b
AB	10001	0000001
AC	10100	0000010
BC	01100	0000100

The corresponding reduced spatial matrices are as follows.

$$RSM_1 = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & 12 & 8 \\ 3 & 0 & 11 \\ 1 & 9 & 0 \end{array}\right] \end{array} \qquad RSM_2 = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & 10 & 13 \\ 11 & 0 & 9 \\ 1 & 3 & 0 \end{array}\right] \end{array}$$

$$RSM_3 = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & 12 & 9 \\ 5 & 0 & 1 \\ 6 & 6 & 0 \end{array}\right] \end{array} \qquad RSM_4 = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} 0 & 4 & 4 \\ 4 & 0 & 8 \\ 7 & 1 & 0 \end{array}\right] \end{array}$$

(1) Following the hash function θ_r, we construct RS of the four pictures.
$RS_1 =$ 1010000010000 0000000100110 10100 10001 01100 10100 01100 10001.
$RS_2 =$ 1010000000100 0000000011001 10100 01100 10001 01100 10001 10100.
$RS_3 =$ 0000110000000 1000000010010 10001 11100 01100 10100 10001.
$RS_4 =$ 1001001000000 0001000100000 01100 10001 10100 10101 01100.

(2) Following the hash function θ_b, we construct BS of the two blocks.
$BS_1 =$ 1010000010100 0000000111111 10100 11101 01100 10001 10100 01100 10001 01100 10001 10100.
$BS_2 =$ 1001111000000 1001000110010 01100 10001 10001 11100 10100 01100 10101 01100 10100 10001.

(3) We use the same algorithm stated above to construct QRS and QBS.

QRS = 0000001000000 0001000000000 10100 10100.

QBS = 0000001000000 0001000000000 0000010 0000010.

(4) Since $QBS^1 \cap BS_1^1 \neq QBS^1$, block 1 will not be selected. And then we check block 2. Since $QBS^1 \cap BS_2^1 = QBS^1$, and $QBS^{2x}(7) \cap BS_2^{2x}(7) = QBS^{2x}(7)$, $QBS^{2y}(4) \cap BS_2^{2y}(4) = QBS^{2y}(4)$, block 2 is selected.

(5) Check the two pictures P_3 and P_4 in block 2 by making use of RS. Only P_4 satisfies $QRS^{2x}(7) \cap RS_4^{2x}(7) = QRS^{2x}(7)$, $QRS^{2y}(4) \cap RS_4^{2y}(4) = QRS^{2y}(4)$. Therefore, P_4 is a possible answer.

(6) Finally, after we compare the RSM of P_4 with the query picture, P_4 matches the query condition.

3 A Prime-Number-Based Access Method

Since in Chang et al.'s fast spatial match access strategy [1], bit-patterns are used as signatures and a bit-wise-or operation is used to union the signatures of the set of object pairs, a false match may occur. Note that the result of a bit-wise-or operation for some signatures may form another signature which is already defined. Therefore, to further reduce the ratio of false match, in [2], Chang proposed a *module-oriented signature extraction* that uses the module operation to filter out the impossible images [2]. Multi-level signature is applied, too. In this strategy, the bit strings used in Chang's fast spatial match access strategy [1] are replaced with prime numbers, the bit-wise-or operation is replaced with a multiply operation, and the bit-wise-and operation in the query processing is replaced with a module operation. Note that the result of a multiplication operation for some prime numbers will never be a prime number again. That is, these two functions θ_b and θ_r map each symbol pair into a prime number, not a bit string. Moreover, in this strategy, θ_r and θ_b are the same.

Following the similar idea, we propose a prime-number-based access method. In the proposed strategy, the data structure of a record signature is almost the same as shown in the first strategy. The only one difference is that RS^2 consists of 26 products of prime numbers instead of 26 bit strings. And $RS^{2x}(i) = \Pi\ \theta(ABi)$, $ABi \in T^x$, $1 \leq i \leq 13$, $RS^{2y}(i) = \Pi\ \theta(ABi)$, $ABi \in T^y$, $1 \leq i \leq 13$. We use the same example shown in Figure 5 in the previous section to show how the algorithm works. The hash function θ is shown as follows.

spatial string	θ
AB	2
AC	3
BC	5

(1) Following the hash function θ, we construct RS of the four pictures.

RS_1 = 1010000010000 0000000100110 3 2 5 3 5 2.

RS_2 = 1010000000100 0000000011001 3 5 2 5 2 3.

RS_3 = 0000110000000 1000000010010 2 15 5 3 2.

RS_4 = 1001001000000 0010000100000 5 2 3 6 5.

(2) Following the hash function θ, we construct BS of the two blocks.

BS_1 = 1010000010100 0000000111111 9 10 5 2 3 5 2 5 2 3.

BS_2 = 1001111000000 1001000110010 5 2 2 15 3 3 6 5 3 2.

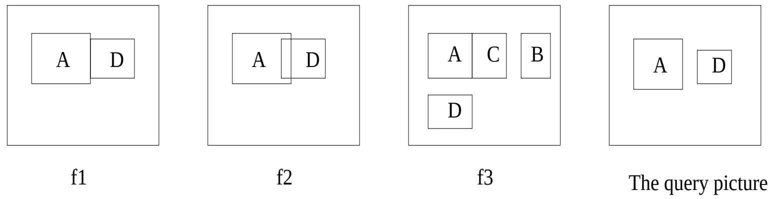

Figure 6: Examples for comparison

(3) We use the same algorithm stated above to construct QRS and QBS.

QRS = 0000001000000 0001000000000 3 3.

QBS = 0000001000000 0001000000000 3 3.

(4) Since $QBS^1 \cap BS_1^1 \neq QBS^1$, block 1 will not be selected. And then we check block 2. Since $QBS^1 \cap BS_2^1 = QBS^1$, and $BS_2^{2x}(7) \bmod QBS^{2x}(7) = 0$, $BS_2^{2y}(4) \bmod QBS^{2y}(4) = 0$, black 2 is selected.

(5) Check the two pictures P_3 and P_4 in block 2 by making use of RS. Only P_4 satisfies $RS_4^{2x}(7) \bmod QRS^{2x}(7) = 0$, $RS_4^{2y}(4) \bmod QRS^{2y}(4) = 0$. Therefore, P_4 is a possible answer.

(6) Finally, after we compare the RSM of P_4 with the query picture, P_4 matches the query condition.

4 A Comparison

The performance of an access method is measured by the ratio of a false match. Obviously, as the size of a signature is increased, the ratio of a false match is decreased. However, the larger the size of a signature is, the more the space needs. Moreover, the design of perfect functions θ_b and θ_r helps to reduce the false match ratio, too. Therefore, the performance of an access method depends on the size of a signature and the hash functions θ_r and θ_b, which has been studied in [2]. In general, the ratio of a false match in prime-number-based algorithms will be smaller than that in bit-string-based algorithms. The reason is that the union of two bit strings may be the same as another bit string, while the product of two prime numbers will never be the same as another prime number.

In our method, we can record 169 spatial relationships between objects, as compared to only 9 spatial relationships represented in other access methods [1, 2, 4]. Due to the same reason, we can distinguish some similar pictorial pictures, while them seem to be the same in other access strategies. Moreover, some pictures with overlapping are different to be handled in 9DLT representation and 2D string representation. For the pictures, f_1 and the query picture, as shown in Figure 6, they have the same 9DLT matrix or 2D string representation, while they have different representations in our access methods. And in this example, f_1 will be retrieved by the query picture using their strategies. Although it is not a false match, it is clear that there are quite a lot of differences between the two pictures. Moreover, an ambiguous case occurs again by using their strategies, when pictures are overlapped, like f_2 shown in Figure 6, where a picture is defined to be *ambiguous* in the representation if there exists more than one different reconstructed picture g from its representation.

Given the same image databases, the same query picture and the same hash functions, the ratio of the false match in our strategy will never be greater than that of Lee et al.'s strategy [4] or Chang et al.'s strategies [1, 2]. Consider picture f_3 and the query picture shown in Figure 6. Assume that $\theta_r(AB) = 1100$, $\theta_r(AC) = 0110$ and $\theta_r(AD) = 1010$. According to Chang et al.'s strategy [1], the record signatures of f_3 and the query picture are constructed as follows: $RS_3^2(7) = \theta_r(AB) \cup \theta_r(AC) = 1110$, and $QRS^2(7) = \theta_r(AD) = 1010$. Since $QRS^2(7) \cap RS_3^2(7) = QRS^2(7)$, picture f_3 will be returned for the query shown in the query picture, i.e., a false match will occur in Chang's strategy [1]. But for this example, a false match will not occur by using our strategy under the same hash functions. According to our strategy, for this example, $RS_3^{2x}(1) = \theta_r(AB)$ $= 1100$, $RS_3^{2x}(3) = \theta_r(AC) = 0110$, $RS_3^{2x}(13) = \theta_r(AD) = 1010$, and $QRS^{2x}(1) = \theta_r(AD) = 1010$. Since $QRS^{2x}(1) \cap RS_3^{2x}(1) \neq QRS^{2x}(1)$, picture f_3 will not be returned.

5 Conclusions

In the previous approaches to retrieve a symbolic picture from an image database which contains multiple symbolic pictures, they only consider 9 spatial relationships that are represented in 2D strings or 9DLT matrices. In this paper, based on the concept of superimposed coding and two-level signature files, we have proposed two access methods which can handle 169 spatial relationships. We have shown that we can distinguish some similar pictorial pictures, while they seem to be the same in other access strategies. Moreover, given the same image databases, the same query picture and the same hash functions, the ratio of the false match in our strategy will never be greater than that of Lee et al.'s strategy or Chang et al.'s strategies.

References

[1] C. C. Chang and J. C. Lin, "A Fast Spatial Match Accessing Scheme for Symbolic Pictures," *Journal of Electrical Engineering,* Vol. 33, No. 3, pp. 129-137, June 1990.

[2] C. C. Chang and H. C. Wu, "A Module-Oriented Signature Extraction to Retrieve Symbolic Pictures," *Journal of Computer,* Vol. 2, No. 4, pp. 45-54, 1990.

[3] C. Faloutsos, "Signature Files: Design and Performance Comparison of Some Signature Extraction Methods." *ACM SIGMOD,* pp. 63-82, May 1985.

[4] S. Y. Lee and M. K. Shan, "Access Methods of Image Database," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 4, No. 1, pp. 27-44, 1990.

[5] S. Y. Lee and F. J. Hsu, "2D C-String: A New Spatial Knowledge Representation for Image Database Systems," *Pattern Recognition,* Vol. 23, No. 10, pp. 1077-1087, 1990.

[6] R. Sacks-Davis and A. Kent, "Multikey Access Methods Based on Superimposed Coding Techniques," *ACM Trans. on Database Systems,* Vol. 12, No. 4, pp. 655-696, 1987.