An Efficient Signature Extraction Method For Image Similarity Retrieval [1]

Wei-Horng Yeh and Ye-In Chang

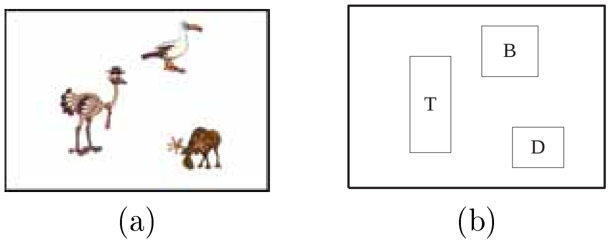
Dept. of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan
Republic of China
{E-mail: changyi@cse.nsysu.edu.tw}
{Tel: 886-7-5252000 (ext. 4334)}
{Fax: 886-7-5254301}

Abstract

The target of similarity retrieval is to retrieve the images that are similar to the query image. Good access methods for large image databases are very important for efficient retrieval. The 2D B-string-based and the unique-ID-based signature methods can provide four kinds of similarity retrieval, object and type-i, $0 \leq i \leq 2$, and can distinguish 169 spatial relationships. However, 169 spatial relationships are still not sufficient to show all kinds of spatial relationships between any two objects in 2D space, for example, the directional relationships, like north. Moreover, in most of the previous methods for similarity retrieval, to simplify the concerns, they apply the MBRs of two objects to define the spatial relationship between them. The topological relationships, however, between objects can be quite different from the spatial relationship of their respective MBRs. Therefore, in this paper, by focusing on the above two problems, we propose a new method. To solve the first problem, we add 9 directional relationships to 169 spatial relationships. In this way, we can distinguish up to 289 spatial relationships in 2D space. To handle the second problem, we adopt the concept of topological relationships in our proposed method. From our simulation study, we show that our method can provide a higher correct match rate than the 2D B-string-based and the unique-ID-based signature methods.

(*Keywords*: 2D strings, access methods, image databases, signatures, similarity retrieval)

[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.



(a) (b)

Figure 1: An example of physical and logical parts: (a) the physical picture; (b) the logical picture.

1 Introduction

Recently, in the field of pattern recognition and image processing, much attention has been paid to the design of image database systems [2, 11, 25]. Applications which use image databases include office automation, computer aided design, robotics, and medical pictorial archiving. In general, an image database can be divided into two parts: the physical and the logical parts. The logical part is used to describe the image features and the secondary information in the original physical pictures. For example, in Figure 1, the logical picture can be regarded as the abstract model of the corresponding physical picture. By searching the logical picture, the corresponding physical one can be retrieved.

In the logical part of an image database, several methods have been introduced. An extended survey about them can be found in [29]. Most of the methods have been classified into the content-based indexing field [24]. The indexes by content are divided into several categories, such as textual, hot spot, color, texture, shape, sketch and the spatial relationships among the elements. For example, the QBIC system [15] supports queries based on image features such as color, shape, texture, and sketch; while the Intelligent Image Database System (IIDS) [7] focuses on the spatial relationships.

Based on previous researches, the spatial relationships used for spatial similarity retrieval can be roughly classified into four approaches: (1) spatial relationships derived from the symbolic projection, (2) directional relationships, (3) topological relationships, and (4) geometry-based spatial relationships [11, 13]. From the viewpoint of the projection along the x- and y-axes, each object can be viewed as surrounding by a Minimum Boundary

Rectangle (MBR), as proposed in the 2D string method [6]. Based on the symbolic projection (i.e., approach 1) of each object along one of the axes, as proposed in Lee *et al.*'s 2D C-string method [20], there are up to 13 spatial relationships in 1D space. The MBR representation is a simple way to represent an object with any shape in an image, which is helpful and efficient for visualization and database browsing [17]. On the other hand, a new method of similarity retrieval by a nine-direction lower-triangular (9DLT) matrix was proposed in [4]. Based upon the variations of 2D strings or the 9DLT matrix, another data structure, a set of *triples*, to represent the spatial relationship between each pair of objects in a picture, was proposed. For each triple, a hashing value is found and stored. Hence, the problem of image matching becomes a problem of matching hashing value sequences [5, 30]. Moreover, in order to solve the ambiguity of MBRs, a method combining the topological and directional relationships to introduce another spatial relationships which are hashed in a hashing table to answer the spatial queries was proposed in [30]. There are some other methods [9, 16, 18, 19, 23, 27, 31].

When there are a large number of images in the image database and each image contains many objects, the processing time for image retrieval is tremendous. Actually, the objects or spatial relationships among objects can be treated as attributes or keywords of a document. Thus, a signature can act as a searching filter to prune (i.e., filter out) most of the unqualified images. Only the records which match the signature need to be examined further for exact query matches. Therefore, to handle large amounts of image databases, several access methods [8, 10, 13, 21] have been proposed.

In [21], based on the 2D B-string method, Lee *et al.* presented a signature method which contains 4 kinds of signatures for object and type-i similarity retrieval. Figure 2 shows some examples of different types of similarity. As compared to Figure 2-(a), Figure 2-(b) only contains the same objects, Figure 2-(c) contains the same objects and have the same spatial category (the *disjoin* category) between objects, which is referred to as *type-0* similarity. Moreover, as compared to Figure 2-(a), Figure 2-(d) satisfies *type-0* similarity and has the same orthogonal relations, which is referred to as *type-1* similarity; while Figure 2-(e) satisfies *type-1* similarity and has the same spatial relationships in x-axis and y-axis, which is referred to as *type-2* similarity. (Note that the difference between Figure 2-(d)

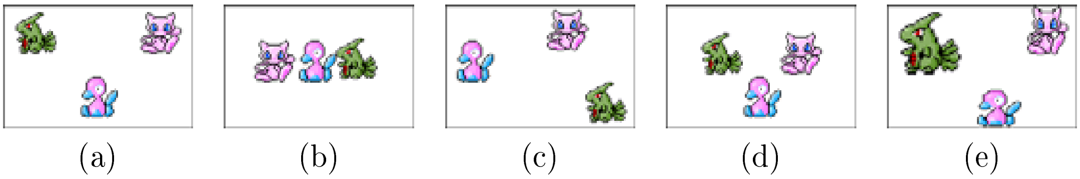


Figure 2: Types of similarity: (a) the original picture; (b) object similarity; (c) type-0 similarity; (d) type-1 similarity; (e) type-2 similarity.

and Figure 2-(e) is that the cat immediately follows, or *meets*, the duck in x-axis in Figure 2-(d); while it is not in Figure 2-(e).) In [10], based on the unique-ID method, Chang *et al.* presented another signature method for object and type-i similarity retrieval. However, Chang *et al.*'s signature contains only type-2 information. Signatures for other type-i and object similarity retrieval are dynamically constructed based on the type-2 signature.

In this paper, two important problems are addressed. One problem is that although the 2D B-string-based [21] and the unique-ID-based [10] signature methods can do similarity retrieval up to distinguish 169 spatial relationships, it is still not sufficient to present all kinds of spatial relationships between any two objects based on 169 spatial relationships. For example, the directional relationship, like south, exists in 2D space and is difficult to be deducted from those 13 spatial operators. Therefore, we propose to consider both of the 9 directional relationships [4] (i.e., approach 2) and the 169 spatial relationships in 2D space. In this way, we can distinguish up to 289 spatial relationships in 2D space. Thus, we have the ability to represent the spatial relationships in 2D space more completely. The other problem is that it is hard to correctly describe the spatial relationships of the objects in terms of relationships between their corresponding *MBR*s. To solve this problem resulted from *MBR*s, we adopting the concept of topological relationships (i.e., approach 3) in our proposed method. In this way, we revise the definition of three kinds of type-i similarity defined in [22] to six kinds to aid similarity retrieval more precisely. From our simulation study, we show that our method can provide a higher correct match rate than the 2D B-string-based and the unique-ID-based signature methods.

The rest of this paper is organized as follows. Section 2 gives a survey of previous proposed representations for symbolic pictures. Section 3 presents the proposed signature

Table 1: Definitions of spatial operators based on the 2D C-string representation

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

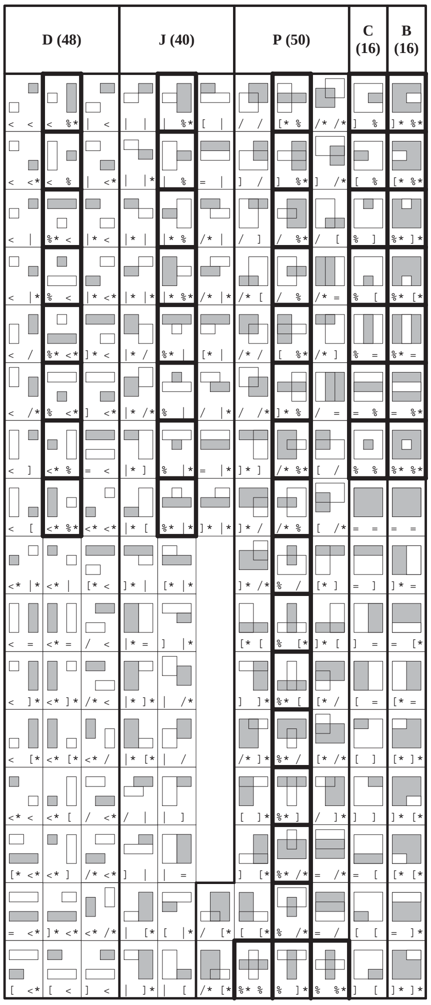
method. Section 4 studies the performance of our proposed method. Finally, Section 5 gives the conclusion.

2 Background

In this section, first, we describe 169 spatial relationships [20], the 9DLT matrix [4], the directional and topological relationships [30] which are major concerns of our proposed method. Next, we describe the structures of the 2D B-string-based signature [21] and the unique-ID-based signature [10].

2.1 Spatial Relationships

Table 1 shows the formal definition of the spatial operators defined in the 2D C-string representation [20]. Those operators and the inverse ones for the related operators represent the spatial relationships between objects in 1D space completely (based on the related positions of the begin bound and the end bound of two objects). Therefore, there are $13 \times 13 = 169$ spatial relationships between two objects in 2D space, as shown in Figure 3 [9], where some of them are surrounded with bold box will be discussed in details in Section 3. (Note that spatial relationships among more than two objects can be represented as the union of spatial relationships between any two of those objects.)

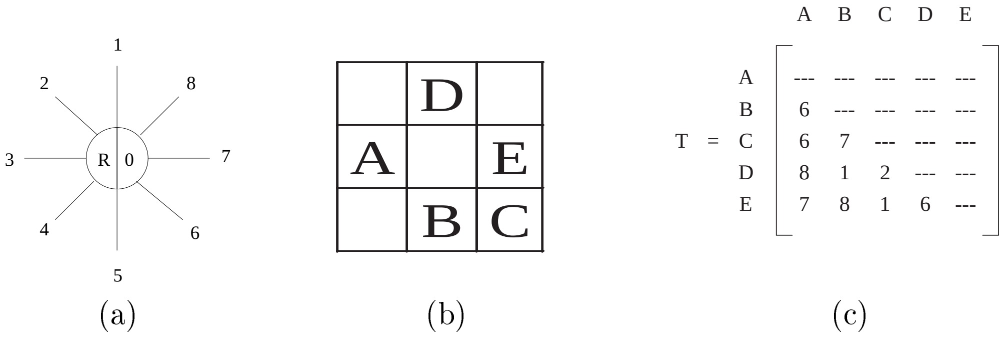


D: Disjoin; J: Join; P: Partial overlap; C: Contain; B: Belong.

Figure 3: 169 spatial relationships in 2D space

For convenience, in [21], they classify 169 spatial relationships into five types of categories: *disjoin*, *join*, *partial overlap*, *contain*, and *belong*. These five categories are called *spatial category relationships* and are described as follows: (1) Disjoin: $A \cap B = \emptyset$; (2) Join: $A \cap B =$ single point or line segment; (3) Contain: $A \cap B = B$; (4) Belong: $A \cap B = A$; (5) Partial overlap: A ∩ B = the area of partial A and partial B. The two objects A and B are enclosed by *MBR*s. The measure criterion for categorization is the area of the intersection of A and B.

They also considered the basic four orthogonal relationships which are *east*, *west*, *north* and *south* [21]. In pictures with no overlapping objects, it is easy to describe the orthogonal



(a)

	D	
A		E
	B	C

(b)

$$T = \begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & --- & --- & --- & --- & --- \\ B & 6 & --- & --- & --- & --- \\ C & 6 & 7 & --- & --- & --- \\ D & 8 & 1 & 2 & --- & --- \\ E & 7 & 8 & 1 & 6 & --- \end{array}$$

(c)

Figure 4: The 9DLT representation: (a) 9 direction codes; (b) a symbolic picture; (c) the related 9DLT matrix.

relationship between objects A and B, because A and B can be regarded as points. But in the case of non-zero sized objects, they described the rules to characterize the orthogonal relationships in the symbolic picture as follows: (1) A is to the east of B iff: $end(A) > end(B)$ on x-axis. (2) A is to the west of B iff: $begin(A) < begin(B)$ on x-axis. (3) A is to the north of B iff: $end(A) > end(B)$ on y-axis. (4) A is to the south of B iff: $begin(A) < begin(B)$ on y-axis.

In [4], C. C. Chang presented the 9 direction codes, as shown in Figure 4-(a). The centroids of two objects are used to obtain the directional relationship between them. For the symbolic picture shown in Figure 4-(b), Figure 4-(c) is the corresponding 9DLT matrix.

In Zhou *et al.*'s method [30], instead of applying the concept of *MBR*s, the directional and topological relationships (i.e., approaches 2 and 3) are combined into one representation. Topological relationships are relationships which are invariant under topological transformation. Given two objects O_i and O_j, their centroids and their boundaries can be used to determine whether the topological relationship between them is *disjoin*, *join*, *contain*, *belong*, or *partial overlap* [12]. (Note that the spatial category relationships mentioned before are different from the topological relationships, even though they use the same terminologies to describe their respective five types.) For example, in Figure 5-(a), the topological relationship between the objects is *disjoin*. But, in Figure 5-(b), the spatial category relationship between the objects of their respective MBRs is *partial overlap*. Zhou *et al.*'s method can avoid this drawback resulted from MBRs. Combining the direc-

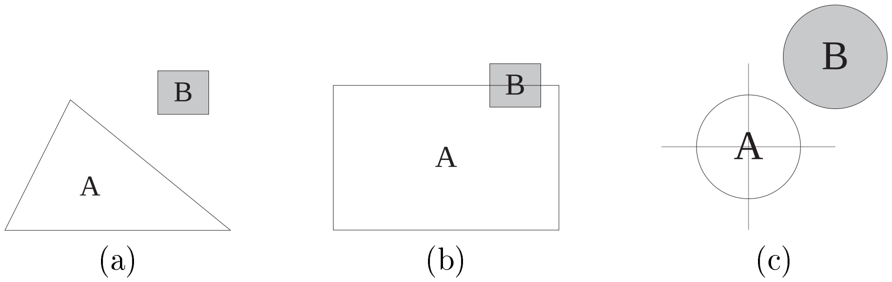


Figure 5: The MBR drawback: (a) the real picture; (b) the related *MBR*s; (c) the directional and topological relationships.

tional and topological relationships, there are 41 types of spatial relationships in 2D space displayed in Figure 6. Let's call this method, the DT (Direction and Topology) method. Following this classification of 41 spatial types, the objects shown in Figure 5-(a) can be represented in Figure 5-(c) correctly.

However, there are some disadvantages in Zhou *et al.*'s method. One is that these 41 spatial relationships cannot describe the spatial relationships between two objects as precisely as the 169 spatial relationships derived from the MBRs, which may also cause the problem of ambiguity. For example, the spatial relationships between the white and gray objects as shown in Figure 7-(a) are different; however, these four different spatial relationships are classified into the same type (type-ID 2) in Zhou *et al.*'s method as shown in Figure 7-(b). The other disadvantage is that we find one missing spatial relationship in Zhou *et al.*'s method. Take Figure 8 for example. Because those two objects shown in Figure 8-(a) have the same centroid, the spatial relationship between them is with type-ID 0 as shown in Figure 8-(b). In fact, the spatial relationship between these two objects should be the one as shown in Figure 8-(c). However, this type of spatial relationship belonging to the partial overlap category does not exist in their proposed 41 spatial relationships.

2.2 Signatures

In [21], Lee *et al.* proposed the 2D B-string-based signature by using superimposed coding and disjoint coding to speed up the access. The integrated signature, as shown in Figure 9, can handle the retrieval by objects, by binary spatial relationship and by subpicture to

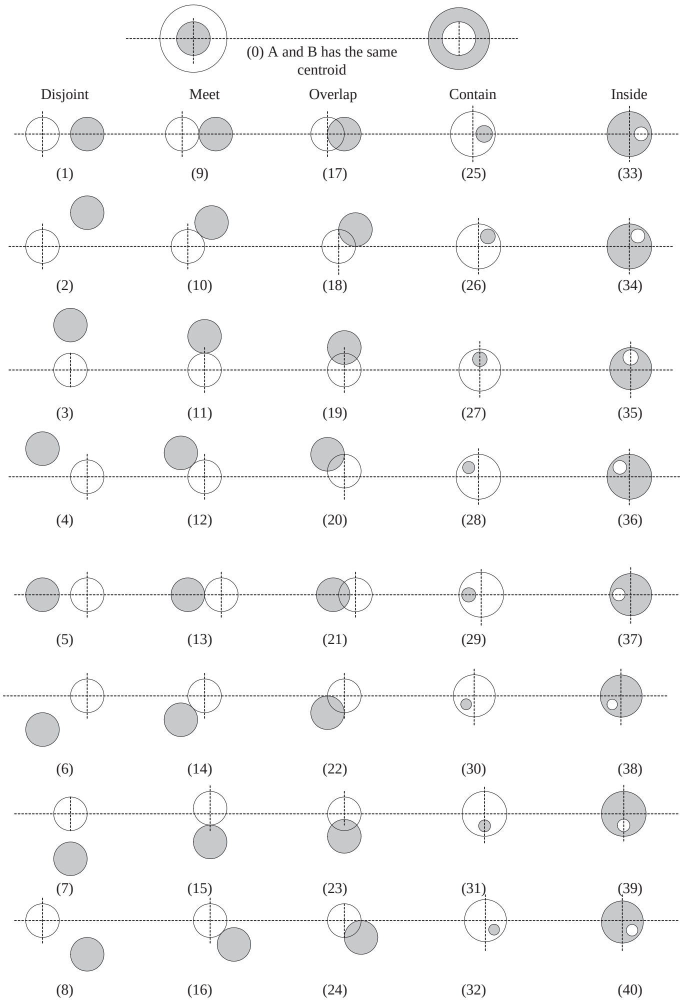


Figure 6: 41 types of spatial relationships in 2D space

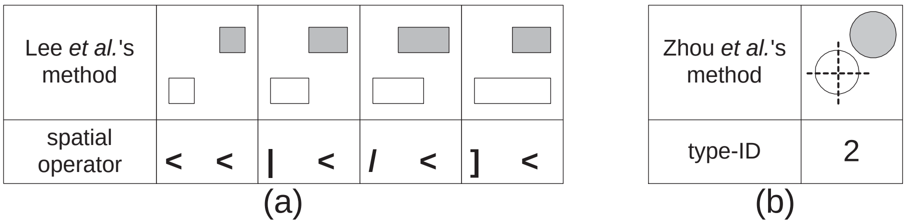


Figure 7: An ambiguous case in Zhou *et al.*'s method: (a) 4 different spatial relationships in Lee *et al.*'s method; (b) one unique code (type-ID 2) in Zhou *et al.*'s method.

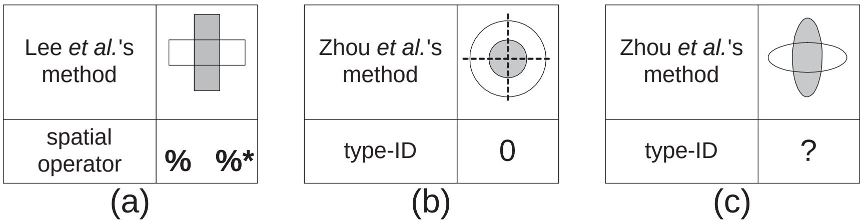


Figure 8: The missing spatial relationship in Zhou *et al.*'s method: (a) the $(\%, \%^*)$ spatial relationship in Lee *et al.*'s method; (b) the spatial relationship with type-ID 0 in Zhou *et al.*'s method; (c) an unknown type-ID (spatial relationship) in Zhou *et al.*'s method.

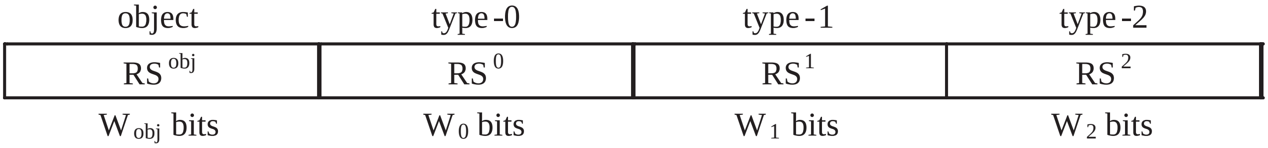


Figure 9: The structure of the 2D B-string-based signature

different types of similarity.

In [10], Chang *et al.* defined a unique-ID-based signature consisting of RS^1 and RS^2, as shown in Figure 10. RS^1 contains RS^{1x} (13-bit string) and RS^{1y}, which represent the record signature flags from the viewpoint of x- and y-axes, respectively. These two 13-bit strings are used to indicate the existence or absence of those 13 spatial operators along the x- and y-axes, respectively. RS^2 contains RS^{2x} (13 bit strings) and RS^{2y}. The i-th bit string 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.

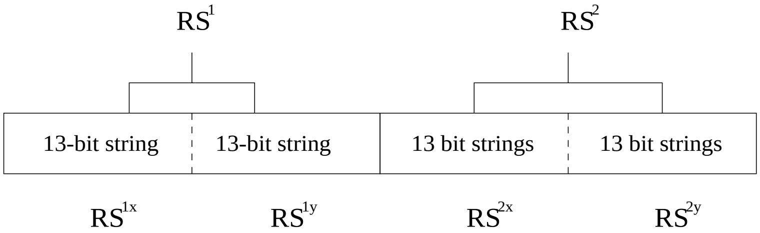


Figure 10: The structure of the unique-ID-based signature

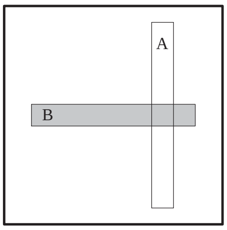


Figure 11: An example of the orthogonal relationship

3 The Proposed Method

From the above observations, we are motivated to find a way to integrate the advantages from Lee *et al.*'s [20] and Zhou *et al.*'s [30] methods (approaches 1, 2, and 3). That is, the 169 spatial relationships, the directional relationships, and the topological relationships, are all considered in our method. We carefully classify each one of 169 spatial relationships into 9 groups based on 9 directional relationships. (Note that the topological relationships between objects are recorded directly in our signature by using a number.) However, some of 169 spatial relationships surrounded with bold box shown in Figure 3 are difficult to be classified.

In this section, we first present the definitions of the extended type-i similarities. Next, we define four new spatial strings. Then, we describe how to construct the record signature. Finally, we present the algorithms for object and type-i similarity retrieval.

3.1 The Extended Type-i Similarities

According to the definition of orthogonal relationships introduced in [21], the situation that both A is to the east of B and A is to the west of B (as shown in Figure 11, for example) may occur. This consideration, however, is not natural to human beings. Thus, similar to the DT method [30], we revise the definition of orthogonal relationships (denoted by $newO$) based on the *centroid* of the iconic objects as follows: (1) A is to the east of B iff: $centroid(A) > centroid(B)$ on x-axis. (2) A is to the west of B iff: $centroid(A) < centroid(B)$ on x-axis. (3) A is to the north of B iff: $centroid(A) > centroid(B)$ on y-axis. (4) A is to the south of B iff: $centroid(A) < centroid(B)$ on y-axis.

From the above discussion, we can extend and revise the existing similarity of types 0, 1, and 2 which were introduced in [21] into types 0, $1'$, 1.5, $2'$, 2.5, and 3 as follows.

Definition 1. *Picture f' is a type-i similar picture 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 $newO_{AB}$ B, A $9D_{AB}$ B, A R_{AB} B, and A T_{AB} B in f
   A C'_{AB} B, A $newO'_{AB}$ B, A $9D'_{AB}$ B, A R'_{AB} B, and A T'_{AB} B in f', then

   type-0: $C'_{AB} = C_{AB}$;

   type-$1'$: *(type-0) and* $(newO'_{AB} = newO_{AB})$;

   type-1.5: *(type-$1'$) and* $(9D'_{AB} = 9D_{AB})$;

   type-$2'$: *(type-$1'$) and* $(R'_{AB} = R_{AB})$;

   type-2.5: *(type-1.5) and (type-$2'$);*

   type-3: *(type-2.5) and* $(T'_{AB} = T_{AB})$;

For the above definition, we have used the following notations: (1) C_{AB} denotes one of the 5 spatial category relationships between A and B; (2) $newO_{AB}$ denotes one of the revised 4 orthogonal relationships between the centroid of A and the centroid of B; (3) $9D_{AB}$ denotes one of 9 direction codes between centroid of A and centroid of B; (4) R_{AB} denotes one of 169 spatial relationships in 2D space between A and B; (5) T_{AB} denotes one of the 5 topological relationships between A and B. For example, in Figure 12, f_0, f'_1, $f_{1.5}$, f'_2, $f_{2.5}$, and f_3 are of type-0, $1'$, 1.5, $2'$, 2.5, and 3 similarity, respectively. Figure 13 shows the hierarchy of the definition for the extended type-i similarity.

There is a division at the *type*-$1'$ similarity. We explain this situation by using two figures as shown in Figures 14 and 15. In Figure 14, since object A in both pictures is to the north east of object B, picture P is of type-1.5 similarity with picture Q. But, the spatial relationship between objects A and B in picture P is "$/^* <^*$", while that in picture Q is "$|^* <^*$". Therefore, picture P is not of type-$2'$ similarity with picture Q, which tells us a fact that pictures are of type-1.5 similarity with each other may not be of type-$2'$ similarity. In Figure 15, object B in both pictures P' and Q' is to the west of object A.

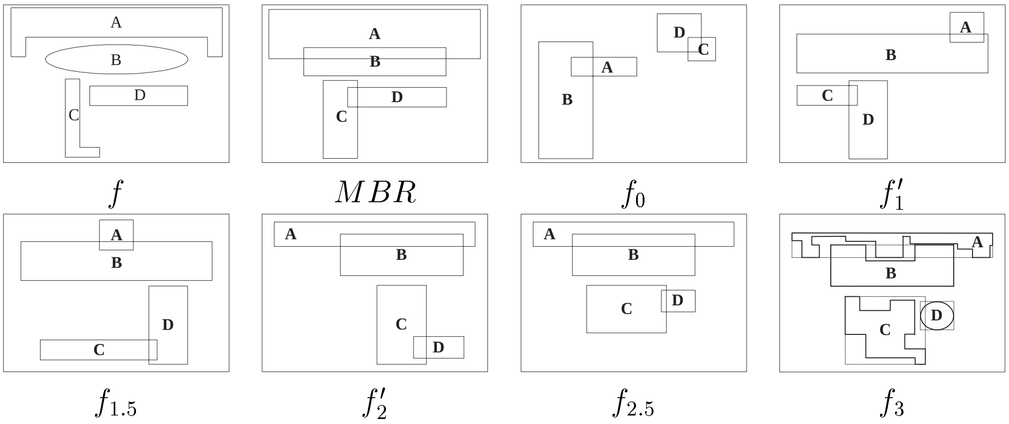


Figure 12: Similarities

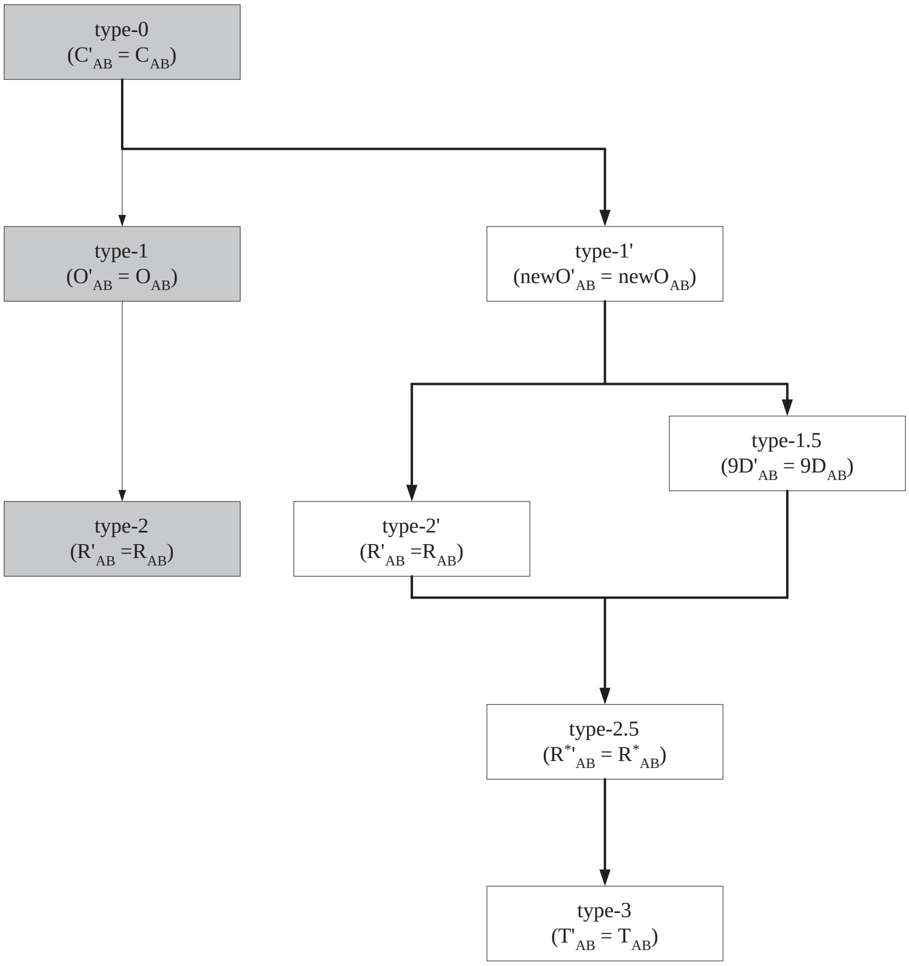


Figure 13: The hierarchy of the type-i similarity

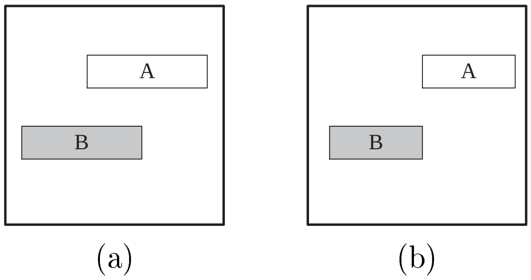


Figure 14: An example of two pictures for type-1.5 similarity: (a) picture P; (b) picture Q.

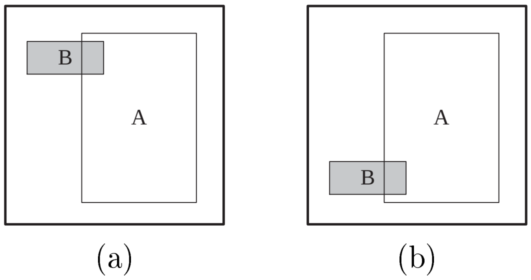
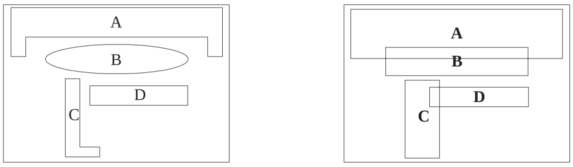


Figure 15: An example of two pictures for type-2$'$ similarity: (a) picture P'; (b) picture Q'.

Moreover, the spatial relationship, "/ %", between them in picture P' is the same as that in picture Q'. Therefore, picture P' is of type-2$'$ similarity with picture Q'. However, object B in picture P' is to the north west of object A, while object B in picture Q' is to the south west of object A. Therefore, Picture P' is not of type-1.5 similarity with picture Q', which tells us another fact that some pictures which are matched with type-2$'$ similarity may not be matched with type-1.5 similarity. Note that although the set of pictures of type-1.5 similarity is not always a subset of pictures of type-2$'$ similarity and vise versa, we still put the position of type-2$'$ similarity lower than that of type-1.5 similarity in the hierarchical picture shown in Figure 13. The reason is that pictures of type-1.5 similarity can only be distinguished into 9 different spatial relationships, while pictures of type-2$'$ similarity can be classified into 169 different spatial relationships.

The branches join together at type-2.5 similarity. The reason is that based on combining 9 directional relationships (type-1.5) with 169 spatial relationships in 2D space (type-2$'$),



(a) (b)

Figure 16: One example of picture f: (a) the real picture, (b) the corresponding *MBR*.

we can induce up to 289 spatial relationships in 2D space which can represent all kinds of spatial relationships more completely than 169 spatial relationships. The symbol R^* shown in Figure 13 is used to represent one of the 289 spatial relationships which will be described later.

3.2 Four New Spatial Strings

To support the representation of the extended type-i similarities, we present four new spatial strings: SCS, DCS, INS, and TRS.

3.2.1 Spatial Category Strings (SCS)

For the picture shown in Figure 16, the corresponding *spatial matrix* S is shown as follows, where the spatial relationship between A and B along x-axis (y-axis) is $A \ /^* \ B$ ($A \ \% \ B$):

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

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

$$RSM = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \left[\begin{array}{cccc} 0 & 6 & 2 & 2 \\ 9 & 0 & 2 & 2 \\ 9 & 9 & 0 & 9 \\ 9 & 7 & 5 & 0 \end{array} \right] \end{array}$$

```
Function CATEGORY($Sid^x_{A,B}$, $Sid^y_{A,B}$)
    if ($Sid^x_{A,B} > 4$) and ($Sid^y_{A,B} > 4$) then
        if ($7 \leq Sid^x_{A,B} \leq 10$) and ($7 \leq Sid^y_{A,B} \leq 10$) then
            return (2) (* contain *)
        else if ($10 \leq Sid^x_{A,B} \leq 13$) and ($10 \leq Sid^y_{A,B} \leq 13$) then
                return (3) (* belong *)
            else return 4 (* partial overlap *)
    else if ($Sid^x_{A,B} > 2$) and ($Sid^y_{A,B} > 2$) then
            return (1) (* join *)
        else return (0) (* disjoin *)
```

Figure 17: The category function

According to a given reduced spatial matrix, we can call function *CATEGORY(T[i,j], T[j,i])* [9] (as shown in Figure 17) to construct the related spatial category matrix, C[i,j].

For the picture shown in Figure 16, the corresponding spatial category matrix C is shown as follows, where 0 and 4 denote the join and partial overlap relationships, respectively.

$$C = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \left[\begin{array}{cccc} - & - & - & - \\ 4 & - & - & - \\ 0 & 0 & - & - \\ 0 & 0 & 4 & - \end{array}\right] \end{array}$$

Based on the spatial category matrix, a *spatial category string SCS* is defined as $\{O_iO_jc_{ij} | c_{ij} \in \{0, 1, \cdots, 4\}\}$, where O_i and O_j are objects. Therefore, the corresponding *SCS* set for the spatial category matrix C is *SCS* = $\{AB4, AC0, AD0, BC0, BD0, CD4\}$.

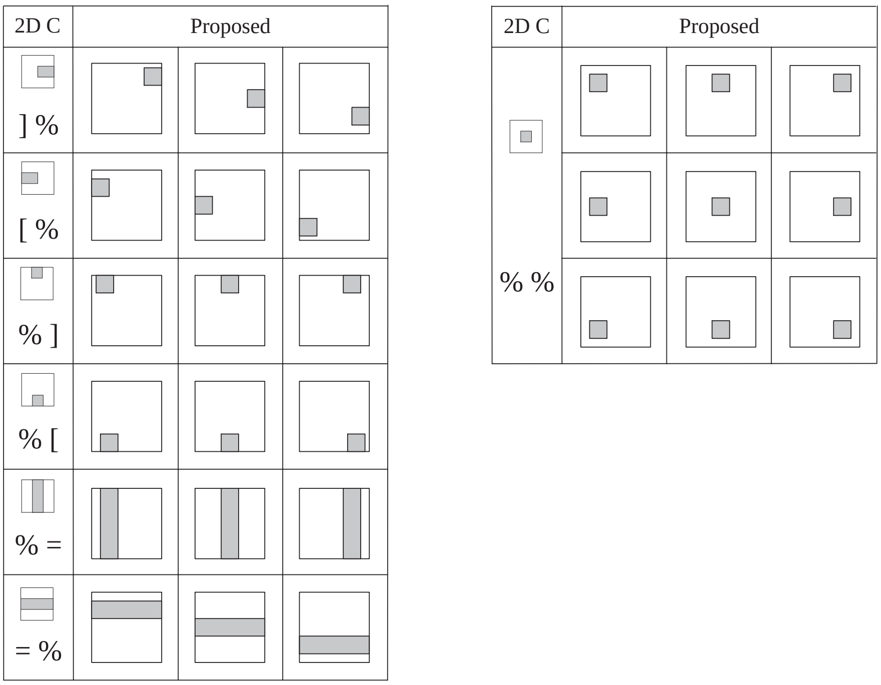
3.2.2 Nine Direction Code Strings (DCS)

According to 9 direction codes, the corresponding 9DLT matrix for Figure 16 is as follows:

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \left[\begin{array}{cccc} - & - & - & - \\ 5 & - & - & - \\ 4 & 4 & - & - \\ 6 & 6 & 8 & - \end{array}\right] \end{array}$$

According to the 9DLT matrix M, a *nine direction code string DCS* is defined as $\{O_iO_jm_{ij} | m_{ij} \in \{0, 1, \cdots, 8\}\}$. Then, the corresponding *DCS* set for Figure 16 is *DCS* = $\{AB5, AC4, AD6, BC4, BD6, CD8\}$.

Table 2: 27 cases in the *contain* category

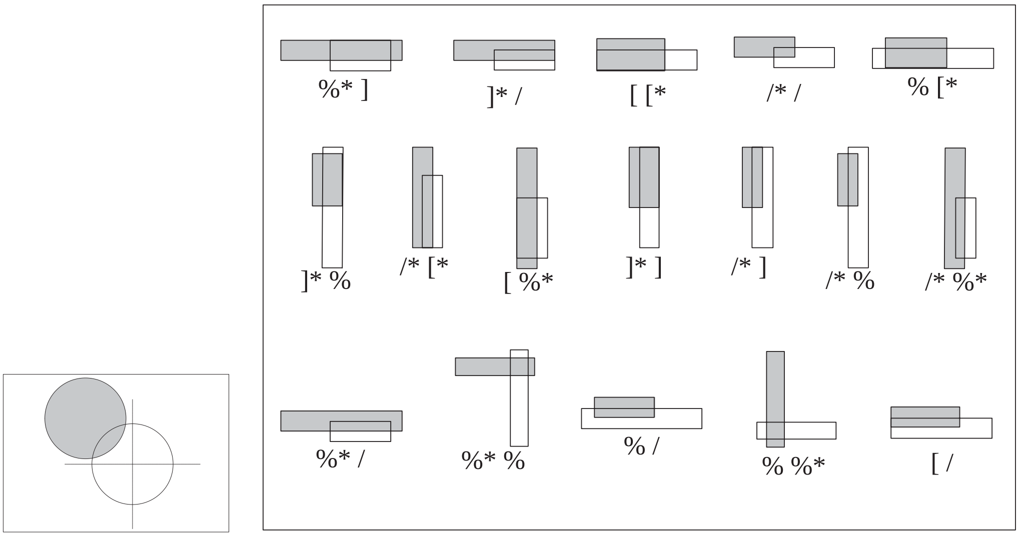


3.2.3 Identification Number Strings (INS)

Based on the definition given in [20], there are 169 spatial relationships in 2D space. However, if we add the information about directional relationships, some spatial relationships among those 169 spatial relationships will be separated into several more kinds of spatial relationships. Take the spatial operator "%%" in Table 2 as an example. Up to 9 cases can occur for the same "%%" operator.

Therefore, the 169 spatial relationships introduced by the 2D C-string method [20] are not sufficient to represent spatial relationships in 2D space. In this way, we preserve those operators used in 169 spatial relationships, then we integrate spatial category relationships and directional relationships into one representation to divide spatial relationships in 2D space. Take Figure 18 as an example. There are 17 spatial relationships which belong to the *partial overlap* spatial category accompanied with the direction code 2 (i.e., *north west*). Note that the number of spatial relationships in each division is variant.

Therefore, after taking 9 directions into consideration, the total number of spatial relationships in our proposed method is 289 ($= 64 + 56 + 36 + 35 + 98$), as summarized in



(a) (b)

Figure 18: One example for the integration of the spatial category and directional relationships: (a) Partial overlap category and the direction code 2; (b) 17 possible spatial relationships.

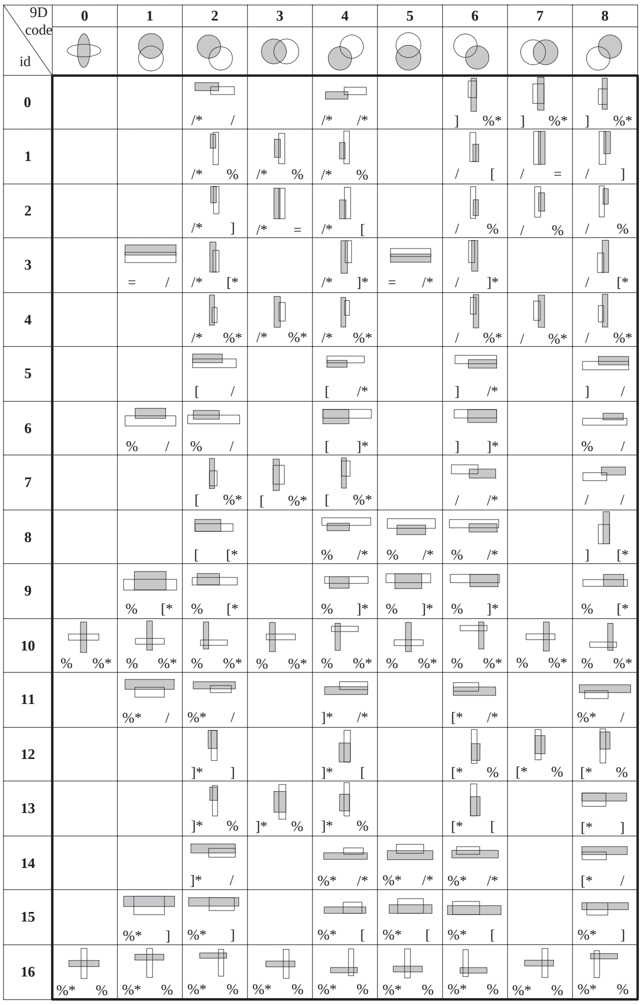
Table 3: The number of spatial relationships in the proposed method

	9 direction codes								
Category	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Disjoin (64)	0	3	13	3	13	3	13	3	13
Join (56)	0	3	11	3	11	3	11	3	11
Contain (36)	4	4	4	4	4	4	4	4	4
Belong (35)	3	4	4	4	4	4	4	4	4
Partial Overlap (98)	2	7	17	7	17	7	17	7	17

Table 3. We assign the identification number to each spatial relationship carefully such that for the same spatial operator with different direction codes is assigned with the same identification number [28]. For example, the same spatial operator "%%*" with different direction codes shown in Table 4 has the same identification number 10. Based on this arrangement, we can distinguish up to 289 spatial relationships which include the original 169 spatial relationships.

Because the information about spatial category relationships and 9 directional relationships are stored in *SCS* and *DCS*, respectively, we just further record the identification number of the related spatial relationship between two objects. Therefore, we define the

Table 4: Identification numbers for the partial overlap spatial category



9D code / id	0	1	2	3	4	5	6	7	8
0			/* /		/* /*		] %*	] %*	] %*
1			/* %	/* %	/* %		/ [	/ =	/]
2			/*]	/* =	/* [		/ %	/ %	/ %
3		= /	/* [*		/*]*	= /*	/]*		/ [*
4			/* %*	/* %*	/* %*		/ %*	/ %*	/ %*
5			[/		[/*		] /*		] /
6		% /	% /		[ ]*		]]*		% /
7			[%*	[%*	[%*		/ /*		/ /
8			[[*		% /*	% /*	% /*		] [*
9		% [*	% [*		%]*	%]*	%]*		% [*
10	% %*	% %*	% %*	% %*	% %*	% %*	% %*	% %*	% %*
11		%* /	%* /		]* /*		[* /*		%* /
12			]*]		]* [		[* %	[* %	[* %
13			]* %	]* %	]* %		[* [		[*]
14			]* /		%* /*	%* /*	%* /*		[* /
15		%*]	%*]		%* [	%* [	%* [		%*]
16	%* %	%* %	%* %	%* %	%* %	%* %	%* %	%* %	%* %

identification number string INS as $\{O_iO_jid_{ij}|id_{ij} \in \{0, 1, \cdots, 16\}\}$. Consequently, the corresponding INS set for the Figure 16 is $INS = \{AB8, AC7, AD7, BC7, BD9, CD2\}$. Therefore, based on the specific combination of the *SCS*, *DCS*, and *INS*, we can discriminate one of 289 spatial relationships between any two objects.

3.2.4 Topological Relationship Strings (TRS)

Basically, a $m \times m$ *Topological Relationship Matrix TRM* of picture f has similar definition as a *Spatial Category Matrix C*, except that the codes of topology relationships are recorded instead of the codes of spatial categories. For the picture shown in Figure 16, the

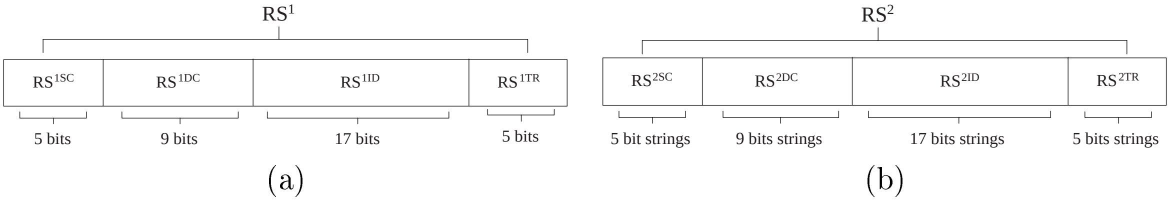


Figure 19: The structure of the proposed signature: (a) RS^1; (b) RS^2.

corresponding topological relationship matrix *TRM* is shown as follows:

$$TRM = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & - & - & - & - \\ B & 0 & - & - & - \\ C & 0 & 0 & - & - \\ D & 0 & 0 & 0 & - \end{array}$$

Then, a *topological relationship string* TRS is defined as $\{O_iO_jt_{ij}|t_{ij} \in \{0, 1, \cdots, 4\}\}$. Therefore, the corresponding *TRS* set for the above topological relationship matrix *TRM* is *TRS* = $\{AB0,\ AC0,\ AD0,\ BD0,\ BD0,\ BC0\}$.

3.3 Record Signature

We now define a Record Signature (*RS*). A *RS* consists of two parts, RS^1 and RS^2 as shown in Figure 19. RS^1 contains four segments, RS^{1SC}, RS^{1DC} ,RS^{1ID}, and RS^{1TR}, which represent the record signature flags to indicate the existence or absence of the numbers representing the meaning defined in each segments. Figure 19 shows the number of bits used in each segment of RS^1. RS^2 consists of four segments, RS^{2SC}, RS^{2DC}, RS^{2ID} and RS^{2TR}. The number of bit strings in each segment is also shown in Figure 19. Table 5 shows each segment notation and corresponding definition, and the algorithm for efficient data access of image databases is described as follows.

Algorithm Record Signature.

(Step 1) List all SCS set, DCS set, INS set, and TRS set.

(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.

Table 5: Notations and related definitions used in record signature segments

Notation	Definition
k_r	the weight (number of 1s) of the record signature
b_r	the length (number of bits) of the record signature
θ_r	the hash function of the record signature
RS_i	the record signature for the ith picture
RS^1	the record signature flags
RS^2	the 36 bit strings of a record signature
RS^{1TR}	the record signature flags from the viewpoint of topological relationship
RS^{1SC}	the record signature flags from the viewpoint of spatial category
RS^{1DC}	the record signature flags from the viewpoint of 9 direction code
RS^{1ID}	the record signature flags from the viewpoint of identification number
RS^{2TR}	the 5 bit strings corresponding to the signature flag field RS^{1TR}
RS^{2SC}	the 5 bit strings corresponding to the signature flag field RS^{1SC}
RS^{2DC}	the 9 bit strings corresponding to the signature flag field RS^{1DC}
RS^{2ID}	the 17 bit strings corresponding to the signature flag field RS^{1ID}
$RS^1(j)$	the jth bit of RS^1
$RS^2(j)$	the jth bit string of RS^2
RS^i_j	the RS^i for the jth picture
QRS	the query record signature

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

(Step 5) For each nine direction code string ABi in *DCS*, we let the i-th bit of RS^{1DC} be 1, and then perform $RS^{2DC}(i) = RS^{2DC}(i) \cup \theta_r(AB)$.

(Step 6) For each identification number string ABi in *INS*, we let the i-th bit of RS^{1ID} be 1, and then perform $RS^{2ID}(i) = RS^{2ID}(i) \cup \theta_r(AB)$.

(Step 7) For each topological relationship string ABi in *TRS*, we let the i-th bit of RS^{1TR} be 1, and then perform $RS^{2TR}(i) = RS^{2TR}(i) \cup \theta_r(AB)$.

(Step 8) Compress RS^2 by removing useless bit strings. If the i-th bit of RS^1 is 0, then remove the corresponding bit-string $RS^2(i)$

To illustrate the algorithm, let's see the following example. For the figure shown in Figure 20, first, we construct the spatial matrix and the reduced spatial matrix (RSM).

$$S = \begin{array}{c|ccc|} & A & B & C \\ A & 0 & /^* & \% \\ B & | & 0 & | \\ C & < & \% & 0 \end{array}$$

$$RSM = \begin{array}{c|ccc|} & A & B & C \\ A & 0 & 6 & 9 \\ B & 3 & 0 & 3 \\ C & 1 & 9 & 0 \end{array}$$

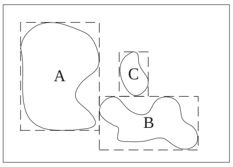


Figure 20: An example

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

1. Generate SCS set, DCS set, TRS set and TRS set. We have $SCS = \{AB1,\ AC0,\ BC1\}$, $DCS = \{AB6,\ AC7,\ BC2\}$, $INS = \{AB1,\ AC10,\ BC0\}$, and $TRS = \{AB0,\ AC0,\ BC0\}$.

2. Design the function θ_r (where $b_r = 5$, $k_r = 2$) which maps each pair of symbols to a unique bit string. For example, we can have $\theta_r(AB) = 10001$, $\theta_r(AC) = 10100$, and $\theta_r(BC) = 01100$

3. Set all bits in RS to 0.
4. If $ABi \in SCS$, we let the i-th bit of RS^{1SC} be 1, and then perform $RS^{2SC}(i) = RS^{2SC}(i) \cup \theta_r(AB)$.
   $RS^{1SC} = 11000$.
   $RS^{2SC}(1) = RS^{2SC}(1) \cup \theta_r(AB) \cup \theta_r(BC) = 11101$.
   $RS^{2SC}(0) = RS^{2SC}(0) \cup \theta_r(AC) = 10100$.

5. Repeat Step 4 by replacing SCS with DCS. We have
   $RS^{1DC} = 001000110$.
   $RS^{2DC}(6) = RS^{2DC}(6) \cup \theta_r(AB) = 10001$.
   $RS^{2DC}(7) = RS^{2DC}(7) \cup \theta_r(AC) = 10100$.
   $RS^{2DC}(2) = RS^{2DC}(2) \cup \theta_r(BC) = 01100$.
6. Repeat Step 4 by replacing SCS with INS. We have
   $RS^{1ID} = 11000000001000000$.
   $RS^{2ID}(1) = RS^{2ID}(1) \cup \theta_r(AB) = 10001$.
   $RS^{2ID}(10) = RS^{2ID}(10) \cup \theta_r(AC) = 10100$.
   $RS^{2ID}(0) = RS^{2ID}(0) \cup \theta_r(BC) = 01100$.
7. Repeat Step 4 by replacing SCS with TRS. We have
   $RS^{1TR} = 10000$.
   $RS^{2TR}(0) = RS^{2TR}(0) \cup \theta_r(AB) \cup \theta_r(AC) \cup \theta_r(BC) = 11101$.
8. Compress RS^2 by removing useless bit strings. If the i-th bit of RS^1 is 0, then remove $RS^2(i)$.

$$\begin{aligned} RS \ = \ & 11000\ 001000110\ 11000000001000000\ 10000 \\ & 10100\ 11101\ 01100\ 10001\ 10100\ 01100\ 10001\ 10100\ 11101 \end{aligned}$$

After the record signature RS and the query signature QRS are constructed, we then can use the condition, $RS \cap QRS \neq QRS$, to decide whether RS should be removed from our consideration.

3.4 Object and Type-i Similarity Retrieval

We will use our proposed structure of a signature discussed in the previous section to do the object and type-i similarity retrieval in the following subsections.

3.4.1 Query of Object Similarity

To simplify our algorithm, we convert each signature back to its *completed* form, instead of the *reduced* form. Given a record signature, we now present an algorithm to convert such a record signature into the related object record signature as follows.

Algorithm Object.
(Convert a Record Signature into an Object Record Signature)

```
(Step 1) Set every bit in $ORS$ to 0.
(Step 2) for i = 0 to 4 do
             if $RS^{1TR}(i) = 1$ then $ORS = ORS \cup RS^{2TR}(i)$.
```

To illustrate the algorithm, let's see the following example. Suppose there are 8 pictures in the database as shown in Figure 21. Let the hash function θ_r have $b_r = 5$ and $k_r = 2$. For example, we can have $\theta_r(AB) = 10001$, $\theta_r(AC) = 10100$, $\theta_r(BC) = 00101$, $\theta_r(AD) = 10010$, $\theta_r(BD) = 00011$, and $\theta_r(CD) = 00110$.

In this case, for example, the corresponding record signatures for picture P_6 is:

RS_6 = 11001 001000100 01100010000000000 11001
10100 10001 00101 00101 10101 10001 10100 00101 10100 10001 00101.

Then, let's see how to convert the record signatures into the object record signatures. Take picture P_6 as an example:

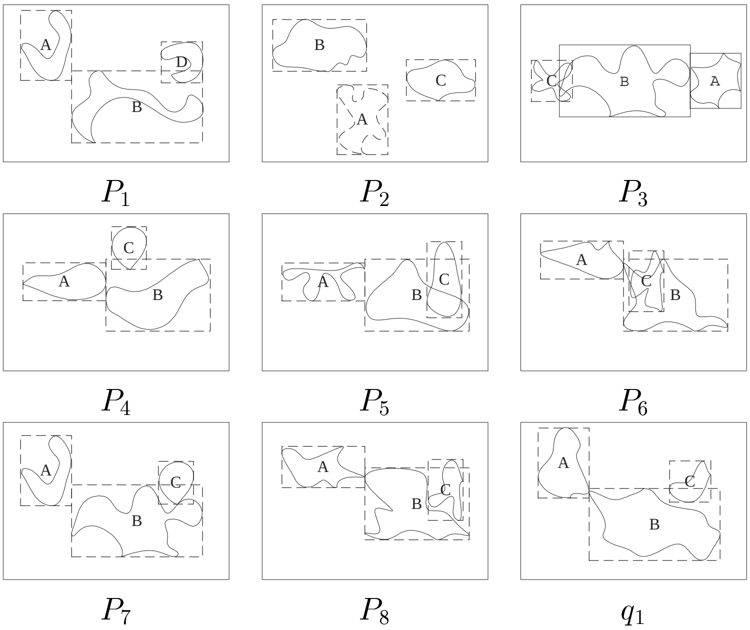
$ORS_6 = 10100 \cup 10001 \cup 00101 = 10101$.

Thus, the object record signatures for pictures P_1 through P_8 are:

$ORS_1 = 10011$, $ORS_2 = 10101$, $ORS_3 = 10101$, $ORS_4 = 10101$,
$ORS_5 = 10101$, $ORS_6 = 10101$, $ORS_7 = 10101$, $ORS_8 = 10101$.

Given a query picture q_1 as shown in Figure 21, the corresponding record signature is

QRS = 11001 000000101 01100010000000000 11000
10100 10001 00101 10101 00101 10001 10100 00101 10101 10001.



P_1 P_2 P_3

P_4 P_5 P_6

P_7 P_8 q_1

Figure 21: An image database $(P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8)$ and a query picture (q_1)

The corresponding object query signature is $QORS = 10101$.

Next, since $QORS \cap ORS_1 \neq QORS$, and $QORS \cap ORS_i = QORS$, $2 \leq i \leq 8$, we conclude that pictures from P_2 through P_8 may have the same objects with the query picture q_1, while picture P_1 has some objects different from the query picture q_1. (Note that, in fact, in the above Algorithm Object, we can replace $RS^{1TR}(i)$ $(RS^{2TR}(i))$ with any of other three fields as shown in Figure 19.)

3.4.2 Query of Type-0 Similarity

The spatial category field (RS^{1SR} and RS^{2SR}) of the record signature contains the information of the spatial category relationship between any two objects in the picture. Since the record signature has the complete information for type-0 similarity (i.e., RS^{1SR} and RS^{2SR}), we only have to compare the related bit-strings to find out the possible answers. Taking the pictures and the query shown in Figure 21 as an example.

1. We have $RS_1^{1SR} \cap QRS^{1SR} = QRS^{1SR}$, $RS_2^{1SR} \cap QRS^{1SR} \neq QRS^{1SR}$, $RS_i^{1SR} \cap QRS^{1SR} = QRS^{1SR}$, $3 \leq i \leq 8$. Picture P_2 needs not to be checked furthermore.

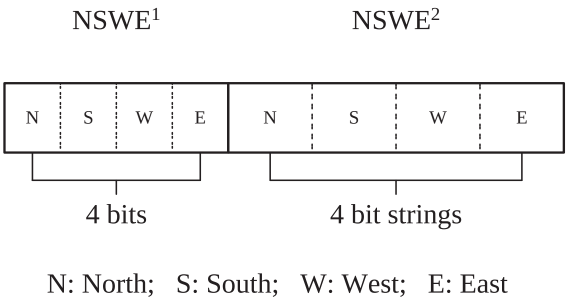


Figure 22: Type-$1'$ signature $NSWE$

2. Next, we have $RS_1^{2SR} \cap QRS^{2SR} \neq QRS^{2SR}$, $RS_3^{2SR} \cap QRS^{2SR} = QRS^{2SR}$, and the remaining pictures have the same result as picture P_3. We conclude that pictures from P_3 through P_8 may be of type-0 similarity with the query picture q_1, while pictures P_1 and P_2 are not of type-0 similarity with the query picture q_1.

3.4.3 Query of Type-$1'$ Similarity

Figure 22 shows the structure of the type-$1'$ record signature *NSWE*. Given a record signature, we now present an algorithm to convert such a record signature into the related type-$1'$ record signature as follows.

Algorithm Type-$1'$.
(Convert a Record Signature into a Type-$1'$ Record Signature)

```
(Step 1) Set every bit in NSWE to 0.
(Step 2) For i ∈ $\{1,2,8\}$ do                                          /* North */
             if $RS^{1DC}(i)$ is 1, set $NSWE^1(0)$ to 1, and let $NSWE^2(0) = NSWE^2(0) \cup RS^{2DC}(i)$.
(Step 3) For i ∈ $\{4,5,6\}$ do                                          /* South */
             if $RS^{1DC}(i)$ is 1, set $NSWE^1(1)$ to 1, and let $NSWE^2(1) = NSWE^2(1) \cup RS^{2DC}(i)$.
(Step 4) For i ∈ $\{2,3,4\}$ do                                          /* West */
             if $RS^{1DC}(i)$ is 1, set $NSWE^1(2)$ to 1, and let $NSWE^2(2) = NSWE^2(2) \cup RS^{2DC}(i)$.
(Step 5) For i ∈ $\{6,7,8\}$ do                                          /* East */
             if $RS^{1DC}(i)$ is 1, set $NSWE^1(3)$ to 1, and let $NSWE^2(3) = NSWE^2(3) \cup RS^{2DC}(i)$.
```

For those 8 pictures from P_1 through P_8 as shown in Figure 21, only pictures from P_3 through P_8 are of type-0 similarity; therefore, we only have to check whether pictures from P_3 through P_8 are of type-$1'$ similarity with the query picture q_1. We now convert the record signatures of pictures from P_3 through P_8 into the type-$1'$ signatures. Take picture P_3 as an example:

```
If $NSWE^1 \cap QNSWE^1 = 0000$ Then
    Return ("the picture is not matched.");
For $i := 0$ to 3 do
Begin
    $j := (i+1) \bmod 4$; $k := (j+1) \bmod 4$;
    flag1 := 0; flag2 := 0; flag3 := 0;
    If $NSWE^1(i) \cap QNSWE^1(i) = 1$ Then
        If $NSWE^2(i) \cap QNSWE^2(i) = \$$ Then flag1 := 1;
    Else flag1 := 2;
    If $NSWE^1(j) \cap QNSWE^1(j) = 1$ Then
        If $NSWE^2(j) \cap QNSWE^2(j) = \$$ Then flag2 := 1;
    Else flag2 := 2;
    If $NSWE^1(k) \cap QNSWE^1(k) = 1$ Then
        If $NSWE^2(k) \cap QNSWE^2(k) = \$$ Then flag3 := 1;
    Else flag3 := 2;
    flag := flag1 * flag2 * flag3;
    If (flag $\neq$ 0 And flag $\neq$ 8) Then Return ("the picture is not matched.");
End;
Return ("two signatures are of type-1$'$ similarity.");
```

† The symbol \$ means that the intersection of two bit strings is not the union of some θ_r.

Figure 23: Algorithm for type-1$'$ similarity retrieval

$$\begin{array}{rcl}
NSWE_3^1(0) & = & NSWE_3^1(1) = NSWE_3^1(3) = 0, \\
NSWE_3^2(0) & = & NSWE_3^2(1) = NSWE_3^2(3) = 00000, \\
NSWE_3^1(2) & = & RS_3^{1DC}(3) = 1, \\
NSWE_3^2(2) & = & RS_3^{2DC}(3) \cup NSWE_3^2(2) = 10101.
\end{array}$$

Thus, $NSWE_3 = 0010\ 00000\ 00000\ 10101\ 00000$.

In the same way, the resulting type-1 record signature for pictures from P_4 through P_7 are as follows:

$$\begin{array}{rcl}
NSWE_4 & = & 1111\ 10101\ 10001\ 00101\ 10101, \\
NSWE_5 & = & NSWE_7 = NSWE_8 = 1101\ 00101\ 10101\ 00000\ 10101, \\
NSWE_6 & = & 1111\ 00101\ 10101\ 00101\ 10101.
\end{array}$$

Given the query picture q_1 as shown in Figure 21, the corresponding type-1$'$ query signature is $QNSWE = 1101\ 00101\ 10101\ 00000\ 10101$.

Next, in Figure 23, we show the algorithm for type-1$'$ similarity retrieval based on the type-1$'$ record signature, *NSWE*.

Therefore, according to the algorithm described in Figure 23, we conclude that from pictures P_4 through P_8 may be of type-1$'$ similarity with the query picture q_1. But, picture P_3 is not of type-1$'$ similarity with the query picture q_1.

3.4.4 Query of Type-1.5 Similarity

The direction code field of the record signature (RS^{1DC} and RS^{2DC}) has the complete information to figure out if two objects has the same directional relationship. Taking advantage of this feature, we can answer the query of type-1.5 similarity retrieval directly.

In Figure 21, only pictures from P_4 through P_8 are of type-$1'$ similarity; therefore, we need only to find out which picture from P_4 through P_8 is of type-1.5 similarity. The steps to check type-1.5 similarity are described as follows.

1. We have $RS_6^{1DC} \cap QRS^{1DC} \neq QRS^{1DC}$ and $RS_j^{1DC} \cap QRS^{1DC} = QRS^{1DC}$, $j \in \{4, 5, 7, 8\}$. Then, picture P_6 is not of type-1.5 similarity with the query picture q_1. Pictures P_4, P_5, P_7, and P_8 need to be checked furthermore.

2. Next, we have $RS_4^{2DC} \cap QRS^{2DC} \neq QRS^{2DC}$. Picture P_4 is not of type-1.5 similarity with the query picture q_1. $RS_j^{2DC} \cap QRS^{2DC} = QRS^{2DC}$, $j \in \{5, 7, 8\}$. So the pictures P_5, P_7, and P_8 may be of type-1.5 similarity with the query picture q_1.

3.4.5 Query of Type-$2'$ Similarity

Before checking the type-$2'$ similarity, we must check the type-$1'$ similarity first. In the previous subsection, "Query of Type-$1'$ Similarity", we have pictures from P_4 through P_8 which are of type-$1'$ similarity with the query picture q_1. To recognize whether two objects have the type-$2'$ similarity or not, we need the information of 169 spatial relationship between them. The identification number field of the record signature (RS^{1ID} and RS^{2ID}) provides with what we need. The steps to find out which picture is of type-$2'$ similarity with the query picture q_1 are stated as follows.

1. Since $RS_i^{1ID} \cap QRS^{1ID} \neq QRS^{1ID}$, $i \in \{4, 5\}$, pictures P_4 and P_5 are not matched and need not to be checked furthermore.

2. Next, since $RS_j^{2ID} \cap QRS^{2ID} = QRS^{2ID}$, $j \in \{6, 7, 8\}$, pictures P_6, P_7 and P_8 may be of type-$2'$ similarity with the query picture q_1.

3.4.6 Query of Type-2.5 Similarity

Suppose the set $S_{1.5}$ contains the pictures which are of type-1.5 similarity with the query picture q_1 and the set $S_{2'}$ contains the pictures which are of type-$2'$ similarity with the query picture q_1. Thus, we have $S_{1.5} = \{P_5, P_7, P_8\}$ and $S_{2'} = \{P_6, P_7, P_8\}$ so far. Based on the definition of type-2.5 similarity, the answer is the intersection of $S_{1.5}$ and $S_{2'}$. That is, pictures P_7 and P_8 are of type-2.5 similarity with the query picture q_1.

3.4.7 Query of Type-3 Similarity

Since pictures P_7 and P_8 are of type-2.5 similarity with the query picture q_1, pictures P_7 and P_8 are the candidates for the type-3 similarity checking. The topological relationship field of the record signature (RS^{1TR} and RS^{2TR}) can help us to do the type-3 similarity retrieval. Thus, the steps are described as follows.

1. Since $RS_i^1 \cap QRS^1 = QRS^1$, $i \in \{7, 8\}$, pictures P_7 and P_8 need to be checked furthermore.
2. Next, we have $RS_7^2 \cap QRS^2 \neq QRS^2$, $RS_8^2 \cap QRS^2 = QRS^2$. Hence, picture P_8 may be matched type-3 similarity with the query picture q_1; but picture P_7 is not.

4 Performance Study

In this section, we compare the performance of the 2D B-string-based [21] and the unique-ID-base [10] signature methods with our proposed method by a simulation study. Moreover, we present the effect of applying the block signature approach to our method.

4.1 A Comparison With the 2D B-String-Based Signature Method

In this comparison, we let the number of different kinds of objects appearing in the database be 60. For each object, the width and height of which are bounded between 1 and 100,000 units. We prepare 2,000 pictures represented in our proposed method and Lee *et al.*'s method in the database in advance. We consider the case of 15 different objects randomly chosen with the uniform distribution to appear in each picture. There are 100 query pictures, where each query picture contains 2 different objects. So, the maximum number

Table 6: A comparison of correct match rates between the 2D B-string-based and our proposed signature methods

	object	type-0	type-1*	type-2*
2D B-based	12.10% (11858/98017)†	8.87% (5403/60902)	3.74% (465/12430)	3.82% (298/7810)
Proposed	80.86% (11858/14665)	40.37% (5403/13383)	30.27% (4045/13363)	5.41% (300/5548)

†(C/S): C is the number of correctly matched pictures, S is the number of pictures passed by the signature.

of matched pictures is 100 × 2,000 = 200,000. The begin bound and end bound on x- and y-coordinates of each object are randomly generated with the uniform distribution. (Note that the data generated in our simulation study could be considered the result of a certain public database pre-processed by a certain image understanding technique which can identify and label objects [13]. For the spatial relationship between any two objects, it can be derived from the coordinates of objects.)

Table 6 shows the comparison of the correct match rate of these two methods. Each rate is calculated from the fractional number below it. The denominator is the number of potential matched pictures judged by Lee *et al.*'s or our methods. The numerator is the number of pictures that actually match the object or type-i similarity with the query pictures. From the denominator shown in Table 6, we observe that our method prunes off more unqualified pictures than Lee *et al.*'s method. The numerator in our method for the object similarity is the same as that in Lee *et al.*'s method, since both methods have the same definition of the object similarity. The denominator in our method is less than that in Lee *et al.*'s method. Thus, our method has a higher correct match rate than Lee *et al.*'s method for the object similarity. Next, because we revise the definition of type-1 similarity in Lee *et al.*'s method as type-1′ similarity, the numerator for type-1 similarity is different from that for type-1′ similarity. The numerator in Lee *et al.*'s method is less than that in our method, which implies that the definition of type-1 similarity is more restrictive than that of type-1′ similarity. (Note that, the correct match rate could be affected by the number of objects in the database and in each picture, and also could be further improved

Table 7: Correct match rates of type-1.5, type-2.5, and type-3 similarity retrieval based on the proposed method

type-1.5	type-2.5	type-3
12% (1334/11120)	4.54% (213/4687)	3.33% (148/4438)

Table 8: A comparison of storage cost

	Min.	Avg.	Max.
2D-B based	2015	2015	2015
Proposed	1411	1523	1631

by choosing the suitable size of a signature and the hash function which has been studied in [3].)

In Table 7, we observe that our method distinguishes more different pictures than Lee *et al.*'s method. For instance, the numbers of matched pictures for type-2.5 and that for type-3 similarity are 213 and 148, respectively. This shows that our method has the ability to do the similarity retrieval precisely based on different criteria.

Next, let's discuss the storage cost of these two methods. In Lee *et al.*'s method, we let the length of the object signature be 15, the total length of the type-0 signature be 50 × 5 (with 50 bits for each category), that of the type-1 signature be 250 × 5, and that of the type-2 signature be 250 × 2, resulting in a total 2015 bits as a record signature for a picture. In our method, we let the length of a bit string be 55, resulting in a total 2016 bits (i.e., 36 bits for RS^1 and 36 × 55 bits for RS^2) as a record signature for a picture. Thus, the size of the signature in these two methods is almost the same.

Table 8 shows the storage cost of these two methods, where "Min.", "Avg.", "Max." stands for the minimum, the average, and the maximum storage cost of one picture, respectively. Note that in our method, if $RS^1(i) = 0$, then $RS^2(i)$ can be removed. Thus, the size of 2016 bits is the upper bound in our method. We observe that the average size of the reduced form of our record signature is smaller than 2016 bits. Moreover, our method needs less storage cost than Lee *et al.*'s method as shown in Table 8.

If we divide the directional relationships into more than 9 cases, for example, one for each degree in a circle, we can support more precise similarity retrieval. However, the storage cost of signatures will be very huge. In this paper, we show that taking 9 directional relationships into consideration is acceptable both in the degree of similarity retrieval and the storage cost.

4.2 A Comparison With the Unique-ID-Based Signature Method

In this comparison, there are 20 different objects and 2,000 pictures in the database. We consider the case that each picture contains 5 different objects. There are 100 query pictures, where each query picture contains 2 different objects. We let each length of RS^{2x} and RS^{2y} bit strings be 100. Thus, the length of the unique-ID-based signature is (13+13) + (13+13) * 100 = 2626. We let the bit-string length of our method be 72. Then, the length of our signature is 2628. (Note that in [10], they have shown that the unique-ID-based signature method outperforms the 2D B-string-based signature method.) A comparison of the correct match rate between the unique-ID-based and our proposed methods is shown in Table 9. (Note that the unique-ID-based method cannot support type-1.5, 2.5 and 3 similarity retrieval.) In Table 9, we show that our method provide a higher correct match rate than the unique-ID-based method. The correct match rate will be affected by many parameters, so those in our method in Tables 6 and 9 are different. For the storage cost, in both methods, if $RS^1(i) = 0$, then $RS^2(i)$ can be removed.

4.3 The Block Signature

The above approach for data filtering in all the methods is based on the kind of the storage organization called *sequential* signatures, as mentioned in $QuickFilter$ [26]. That is, given NR records, we have to compare signatures for NR times sequentially. To reduce the number of comparisons with each record signature in image databases, we can use techniques of block signatures [8, 10], multi-level signatures [13], or dynamic hashing [26] to each of methods. Here, we show the simulation results of applying the block signature (BS) to our method. The algorithm to construct BS is almost the same as RS. The only one difference between them is that we use another hashing function θ_b according to

Table 9: A comparison of correct match rates between the unique-ID-based (UID-based) and our proposed signature methods

	object	type-0	type-1*	type-2*
UID-based	51.76% (10549/20382)	42.51% (4974/11701)	23.28% (1383/5941)	20.98% (292/1392)
Proposed	51.79% (10549/20370)	65.01% (4974/7651)	72.35% (3774/5216)	49.48% (1229/1491)

Table 10: Percentage of signature comparisons for different number of objects per picture

	object	type-0	type-1’	type-1.5	type 2’	type-2.5	type-3
5 objects	31.52%	24.58%	23.78%	15.25%	10.83%	9.80%	9.71%
10 objects	47.40%	43.74%	43.68%	37.08%	23.45%	21.30%	20.80%
15 objects	67.37%	64.09%	65.06%	60.50%	44.16%	41.75%	40.80%

the given k_b (the number of 1’s of the block signature) and b_b (the number of bits of the block signature) to get the block signatures of object blocks. Moreover, the size of the bits generated by the hashing function θ_b is usually larger than that used in a record signature such that we can increase the correct match rate of a block signature.

Let’s consider three cases of 5, 10, and 15 objects in each picture, respectively. There are 2000 pictures and 2000 record signatures in the database. Each block signature records the information of 10 pictures. Thus, we have 2000 / 10 = 200 block signatures and totally 2,200 signatures in the database. From the simulation result shown in Table 10, we observe that the fewer objects are there per picture, the smaller percentage of signature comparisons needed, where the percentage of signature comparisons is equal to (the total number of compared block and record signatures) / (the total number of block and record signatures). This is because a smaller number of objects per picture implies a smaller number of bit 1’s in the block signature and, hence, a smaller number of comparisons.

On the other hand, let’s consider three cases of a block signature recording the information of 5, 10, and 20 records, respectively. Therefore, there are 400 (= 2000 / 5), 200 (= 2000 / 20), and 100 (= 2000 / 20) block signatures in the database, respectively. The total number of signatures in the database for each case is 2400, 2200, and 2100, respectively.

Table 11: Percentage of signature comparisons for different number of records per block

	object	type-0	type-1'	type-1.5	type 2'	type-2.5	type-3
5 records	30.54%	28.91%	28.86%	25.90%	21.00%	20.07%	19.84%
10 records	47.40%	43.74%	43.68%	37.08%	23.45%	21.30%	20.80%
20 records	77.46%	72.42%	72.36%	65.25%	41.03%	38.23%	37.10%

Table 12: Match rate of block signatures for different number of objects per picture

	object	type-0	type-1'	type-1.5	type 2'	type-2.5	type-3
5 objects	24.68%	17.04%	16.16%	6.78%	1.91%	0.78%	0.69%
10 objects	42.13%	38.12%	38.05%	30.79%	15.80%	13.44%	12.88%
15 objects	64.11%	61.59%	61.57%	56.56%	38.58%	35.93%	34.88%

There are 10 objects in each picture. From the simulation result shown in Table 11, we observe that the more records are recorded in one block signature, the larger percentage of signature comparisons needed. This is because a larger number of records recorded in one block signature implies a larger number of matched block signatures ($QBS \cap BS = QBS$) and, hence, a larger number of comparisons, where QBS is the query block signature.

Basically, given NR records, the number of records recorded in one block signature denoted by $NRperB$, and the match rate of block signatures denoted by MRB, the total number of comparisons of record and block signatures (TNC) is equal to
$(NR/NRperB) + MRB * (NR/NRperB) * NRperB =$
$(NR/NRperB) * (1 + MRB * NRperB)$.

Therefore, TNC will be affected by MRB and $NRperB$, given the same NR records. As shown in Table 12, the number of objects per picture will affect MRB, resulting in affecting TNC. As shown in Table 13, the number of records per block, i.e., $NRperB$, will affect MRB and TNC, too. Moreover, the *locality* of objects in records (pictures) and queries will also affect MRB. For example, given 1000 records, 10 objects and a block signature containing the information of every 10 records, let's consider the case in which objects A and B are located in records 1 to 10 only, and most of the queries inquire the relationship between objects A and B. In such a case, MRB will also be reduced,

Table 13: Match rate of block signatures for different number of records per block

	object	type-0	type-1'	type-1.5	type 2'	type-2.5	type-3
5 records	16.65%	14.69%	14.63%	11.08%	5.20%	4.08%	3.81%
10 records	42.13%	38.12%	38.05%	30.79%	15.80%	13.44%	12.88%
20 records	76.33%	71.04%	70.98%	63.15%	38.08%	35.14%	33.96%

resulting in a small value of TNC. In each of the above cases, as long as MRB is smaller than $((NRperB - 1)/NRperB)$, the method of applying the block signatures always needs smaller number of comparisons than the method of sequential signatures.

5 Conclusion

In this paper, we have presented a new method which combines the advantages of the previous methods, the 2D C-string, the 9DLT matrix, and the DT method for similarity retrieval from a large image database. We have extended the existing three kinds of type-i similarity up to six to facilitate similarity retrieval with high accuracy. By adding 9 directions to 169 spatial relationships, we have shown that up to 289 spatial relationships can be used to represent the spatial relationships in 2D space, which can distinguish some spatial relationships that can not be distinguished based on 169 spatial relationships defined in 2D C-string. Moreover, in order to overcome the ambiguity resulted from enclosing symbolic objects by MBRs, we have adopted the concept of topological relationships. Based on the above extensions, we have proposed a new structure of a signature and algorithms to do the object and six type-i similarities. From our simulation results, we have shown that our proposed method provides a higher correct match rate than the 2D B-string-based and the unique-ID-based signature methods. How to handle case of similarity retrieval for images which may be rotated is our future work.

References

[1] A. F. Abate, M. Nappi, G. Tortora, and M. Tucci, "IME: An Image Management Environment with Content-Based Access," *Image and Vision Computing,* Vo. 17, No. 13, pp. 967–980, Nov. 1999.

[2] I. Ahmad and W. I. Grosky, "Indexing and Retrieval of Images by Spatial Constraints," *Journal of Visual Communication and Image Representation,* Vol. 14, No. 3, pp. 291–320, Sep. 2003.

[3] 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.

[4] C. C. Chang, "Spatial Match Retrieval of Symbolic Pictures," *Journal of Information Science and Engineering,* Vol. 7, No. 3, pp. 405–422, Sept. 1991.

[5] C. C. Chang and C. F. Lee, "A Spatial Match Retrieval Mechanism for Symbolic Pictures," *The Journal of Systems and Software,* Vol. 44, No. 1, pp. 73–83, Dec. 1998.

[6] S. K. Chang, Q. Y. Shi and C. W. Yan, "Iconic Indexing by 2D Strings," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, No. 3, pp. 413–428, May 1987.

[7] S. K. Chang, E. Jungert and G. Tortora, *Intelligent Image Database Systems,* World Scientific Press, Singapore, 1996.

[8] Y. I. Chang and B. Y. Yang, "Efficient Access Methods for Image Databases," *Information Processing Letters,* Vol. 64, No. 2, pp. 95–105, Oct. 1997.

[9] Y. I. Chang, H. Y. Ann and W. H. Yeh, "A Unique-Id-Based Matrix Strategy for Efficient Iconic Indexing of Symbolic Pictures," *Pattern Recognition,* Vol. 33, No. 8, pp. 1263–1276, Aug. 2000.

[10] Y. I. Chang, H. Y.Ann and W. H. Yeh, "An Efficient Signature File Strategy for Similarity Retrieval from Large Iconic Image Databases," *Journal of Visual Languages and Computing,* Vol. 13, No. 2, pp. 117–147, April 2002.

[11] E. DiSciascio, M. Mongiello, F. M. Donini, and L. Allegretti, "Retrieval by Spatial Similarity: An Algorithm and a Comparative Evaluation," *Pattern Recognition Letters,* Vol. 25, No. 14, pp. 1633-1645, Oct. 2004.

[12] M. J. Egenhofer, "Point-Set Topological Spatial Relations," *Int. J. Geographical Information Systems,* Vol. 5, No. 2, pp. 161–174, 1991.

[13] E. A. El-kwae and M. R. Kabuka, "Efficient Content-Based Indexing of Large Image Databases," *ACM Trans. on Information Systems,* Vol. 18, No. 2, pp. 171–210, Apr. 2000.

[14] C. Faloutsos and S. Christodoulakis, "Description and Performance Analysis of Signature File Methods for Office Filing", *ACM Trans. on Office Information Systems,* Vol. 5, No. 3, 1987.

[15] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE The Computer Maganize,* Vol. 28, No. 9, pp. 23–32, Sept. 1995.

[16] D. S. Guru and P. Nagabhushan, "Triangular Spatial Relationship: A New Approach for Spatial Knowledge Representation," *Pattern Recognition Letters,* Vol. 22, No. 9, pp. 999–1006, July 2001.

[17] P. W. Huang and C. H. Lee, "Image Database Design Based on 9D-SPA Representation for Spatial Relations," *IEEE Trans. on Knowledge and Data Engineering,* Vol. 16, No. 12, pp. 1486–1496, Dec. 2004.

[18] J. S. Jin and R. Kurniawati, "Varying Similarity Metrics in Visual Information Retrieval," *Pattern Recognition Letters,* Vol. 22, No. 5, pp. 583–592, April 2001.

[19] J. T. Lee and H. P. Chiu, "2D Z-string: A New Spatial Knowledge Representation for Image Databases," *Pattern Recognition Letters,* Vol. 24, No. 16, pp. 3015–3026, Dec. 2003.

[20] 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, Oct. 1990.

[21] S. Y. Lee, M. C. Yang and J. W. Chen, "Signature File as a Spatial Filter for Iconic Image Database," *Journal of Visual Languages and Computing,* Vol. 3, pp. 373–397, 1992.

[22] S. Y. Lee, M. C. Yang and J. W. Chen, "2D B-String: A Spatial Knowledge Representation for Image Database Systems," *Proc. ICSC'92 Second Int. Computer Science Conf.,* pp. 609–615, 1992.

[23] D. C. Lou and T. L. Yin, "Spatial Database With Each Picture Self-Contained Multiscape and Access Control in a Hierarchy," *The Journal of Systems And Software,* Vol. 56, No. 2, pp. 153–163, March 2001.

[24] P. Maresca, A. Guercio, T. Arndt and G. Tortora, "Multimedia Indexing with the SMART System," *Journal of Visual Languages and Computing,* Vol. 11, No. 4, pp. 405–438, Aug. 2000.

[25] E. G. M. Petrakis, "Design and Evaluation of Spatial Similarity Approaches for Image Retrieval," *Image and Vision Computing,* Vol. 20, No. 1, pp. 59–76, Jan. 2002.

[26] F. Rabitti and P. Zezula, "A Dynamic Signature Technique for Multimedia Databases," *Proc. of the 13th Annual Int. ACM/SIGIR Conf. on Research and Development in Information Retrieval,* pp. 193–210, 1989.

[27] Y. H. Wang, "Image Indexing and Similarity Retrieval Based on Spatial Relationship Model,"*Information Sciences,* Vol. 154, No. 1-2, pp. 39–58, Aug. 2003.

[28] W. H. Yeh, "A Hybrid Approach-Based Signature Extraction Method for Similarity Retrieval," *Master Thesis*, Department of Computer Science and Engineering, National Sun Yat-Sen University, June 2001.

[29] A. Yoshitaka and T. Ichikawa, "A Survey on Content-Based Retrieval for Multimedia Databases," *IEEE Trans. on Knowledge and Data Engineering,* Vol. 11, No. 1, pp. 81–93, Jan.–Feb. 1999.

[30] X. M. Zhou and C. H. Ang, "Retrieving Similar Pictures from a Pictorial Database by an Improved Hashing Table," *Pattern Recognition Letters,* Vol. 18, No. 8, pp. 751–758, Aug. 1997.

[31] X. M. Zhou, C. H. Ang, and T. W. Ling, "Image Retrieval Based on Object's Orientation Spatial Relationship," *Pattern Recognition Letters,* Vol. 22, No. 5, pp. 469–477, April 2001.