JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 22, 63-94 (2006)

An Efficient Signature Extraction Method for Image Similarity Retrieval*

WEI-HORNG YEH AND YE-IN CHANG
Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, 804 Taiwan
E-mail: changyi@cse.nsysu.edu.tw

The goal of similarity retrieval is to retrieve images that are similar to query image. Good access methods for large image databases are very important for efficient retrieval. The 2D B-string-based and unique-ID-based signature methods can provide four kinds of similarity retrieval, object and type-*i*, where $0 \le i \le 2$, 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, such as directional relationships, like north. Moreover, most of the previous similarity retrieval methods, for the sake of simplicity, apply the *MBR*s of two objects to define the spatial relationship between them. The topological relationships, however, between objects can be quite different from the spatial relationship between their respective *MBR*s. Therefore, in this paper, we propose a new method that focuses on the above two problems. To solve the first problem, we add 9 directional relationships to the 169 spatial relationships. In this way, we can distinguish up to 289 spatial relationships in 2D space. To handle the second problem, we apply the concept of topological relationships in our proposed method. Based on the results of our simulation study, we show that our method can achieve a higher correct match rate than the 2D B-string-based and unique-ID-based signature methods can.

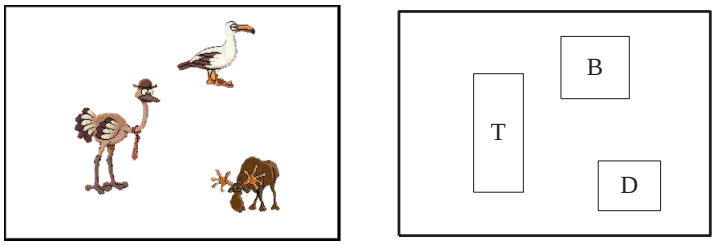
Keywords: 2D strings, access methods, image databases, signatures, similarity retrieval

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 a physical and a logical part. The logical part is used to describe the image features and the secondary information in the original physical pictures. For example, in Fig. 1, the logical picture can be regarded as the abstract model of the corresponding physical picture. When the logical picture is searched, the corresponding physical picture can be retrieved.

For the logical part of an image database, several methods have been introduced. An extended survey of them can be found in [29]. Most of these methods adopt content-based

Received August 5, 2003; revised October 7, 2004 & February 15, 2005; accepted August 15, 2005.
Communicated by Suh-Yin Lee.
* This research was supported in part by the National Science Council of Taiwan, R.O.C., under grant No. NSC 87-2213-E-110-014.



(a) The physical picture. (b) The logical picture.

Fig. 1. An example of physical and logical parts.

indexing [24]. Indexes by based on 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; meanwhile, the Intelligent Image Database System (IIDS) [7] focuses on spatial relationships.

Based on previous researches, the spatial relationships used in spatial similarity retrieval can be roughly classified into four types: (1) spatial relationships derived through 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 being surrounded by a Minimum Boundary Rectangle (*MBR*), as in the 2D string method [6]. Based on the symbolic projection (i.e., approach 1) of each object along one of the axes, as 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 similarity retrieval method employing 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, consisting of a set of *triples* and used 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 that of matching hashing value sequences [5, 30]. Moreover, in [30], in order to deal with the ambiguity of *MBR*s, a method was proposed that combines the topological and directional relationships and uses them to introduce other spatial relationships which are hashed in a hashing table in order to answer spatial queries. Other methods were proposed in [9, 16, 18, 19, 23, 27, 31].

When there are a large number of images in an image database and each image contains many objects, the amount of processing time needed for image retrieval is huge. Actually, the objects or spatial relationships among objects can be treated as the 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 numbers 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. Fig. 2 shows

(a) The original picture. (b) Object similarity. (c) Type-0 similarity.

(d) Type-1 similarity. (e) Type-2 similarity.

Fig. 2. Types of similarity.

some examples of different types of similarity. Compared with Fig. 2 (a), Fig. 2 (b) only contains the same objects, while Fig. 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, compared with Fig. 2 (a), Fig. 2 (d) satisfies *type*-0 similarity and has the same orthogonal relations, which is referred to as *type*-1 similarity, while Fig. 2 (e) satisfies *type*-1 similarity and has the same spatial relationships in the *x*-axis and *y*-axis, which is referred to as *type*-2 similarity. (Note that the difference between Figs. 2 (d) and (e) is that the cat immediately follows, or *meets*, the duck in the *x*-axis in Fig. 2 (d), while it does not in Fig. 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 kinds of type-*i* and object similarity retrieval are dynamically constructed based on type-2 signatures.

In this paper, two important problems are addressed. One problem is that, although the 2D B-string-based [21] and unique-ID-based [10] signature methods can perform similarity retrieval with up to 169 spatial relationships, this is still not sufficient to present all kinds of spatial relationships between any two objects. For example, directional relationships, like south, exists in 2D space and are difficult to deduce 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 spatial relationships in 2D space more completely. The other problem is that it is hard to correctly describe the spatial relationships of objects in terms of the relationships between their corresponding *MBR*s. To solve this problem, we apply the concept of topological relationships (i.e., approach 3) in our proposed method. In this way, we revise the definitions of three kinds of type-*i* similarity defined in [22] and define six kinds similarity in order to make similarity retrieval more precise. From the results of our simulation study, we show that our method can achieve a higher correct match rate than the 2D B-string- based and unique-ID-based signature methods.

The rest of this paper is organized as follows. Section 2 can surveys previously proposed representations for symbolic pictures. Section 3 presents the proposed signature method. Section 4 studies the performance of our proposed method. Finally, section 5 draws conclusions.

2. BACKGROUND

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

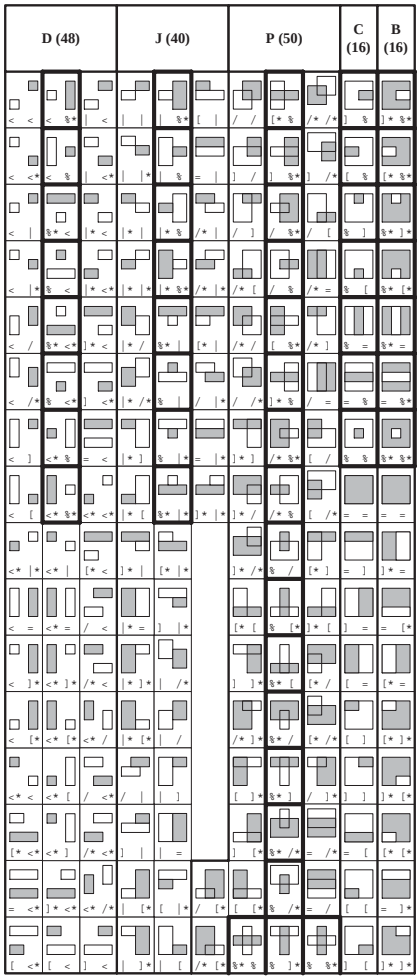
2.1 Spatial Relationships

Table 1 shows formal definitions 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 Fig. 3 [9]. Some of them are enclosed in bold boxes and will be discussed in detail in section 3. (Note that spatial relationships among more than two objects can be represented as the union of the spatial relationships between any two of those objects.)

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 do not have 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 partly overlaps B

For convenience, in [21], the authors divided the 169 spatial relationships into five types: *disjoin*, *join*, *partial overlap*, *contain*, and *belong*. These five types are called *spatial category relationships* and are defined 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: $A \cap B = A$;



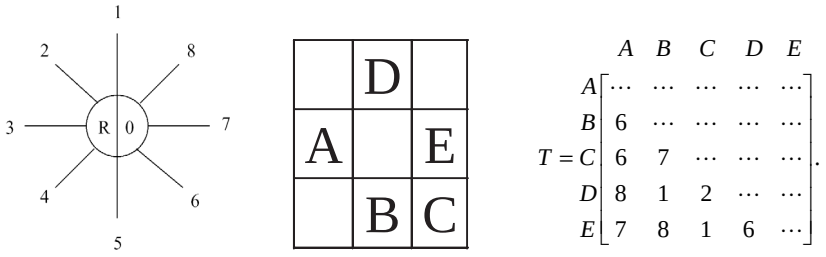
D: disjoin J: join P: partial overlap C: contain B: belong

Fig. 3. 169 spatial relationships in 2D space.

(5) partial overlap: $A \cap B$ = 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 intersection between *A* and *B*.

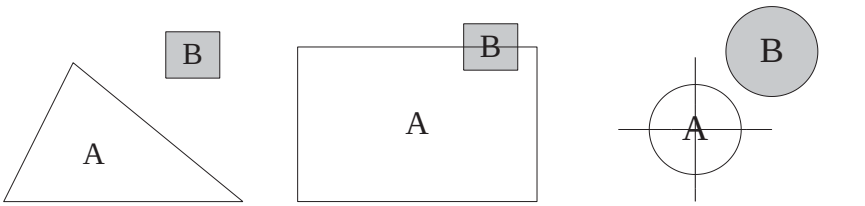
The authors also considered the basic four orthogonal relationships, which are *east*, *west*, *north* and *south* [21]. For pictures with no overlapping objects, it is easy to describe the orthogonal relationship between objects *A* and *B*, because *A* and *B* can be regarded as points. But in the case of non-zero sized objects, the authors described the rules characterizing the orthogonal relationships in the symbolic picture as follows: (1) *A* is to the east of *B* iff: *end*(*A*) > *end*(*B*) on the *x*-axis. (2) *A* is to the west of *B* iff: *begin*(*A*) < *begin*(*B*) on the *x*-axis. (3) *A* is to the north of *B* iff: *end*(*A*) > *end*(*B*) on the *y*-axis. (4) *A* is to the south of *B* iff: *begin*(*A*) < *begin*(*B*) on the *y*-axis.

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



(a) 9 direction codes. (b) A symbolic picture. (c) The related 9DLT matrix.

Fig. 4. The 9DLT representation.



(a) The real picture. (b) The related *MBR*s. (c) The directional and topological relationships.

Fig. 5. The *MBR* drawback.

Zhou *et al.*'s method [30], instead of applying the concept of *MBR*s, combines the directional and topological relationships (i.e., approaches 2 and 3) into one representation. Topological relationships are relationships which are invariant under topological transformation. Given two objects, O_i and O_j, their centroids and 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 previously are different from the topological relationships, even though the same terms are used to describe their respective five types.) For example, in Fig. 5 (a), the topological relationship between the objects is *disjoin*, while in Fig. 5 (b), the spatial category relationship between the objects of their respective *MBR*s is *partial overlap*. Zhou *et al.*'s method can avoid this drawback resulting from *MBR*s. With the directional and topological relationships combined, there are 41 types of spatial relationships in 2D space, as displayed in Fig. 6. Let us call this method the *DT* (Direction and Topology) method. Based on this classification of 41 spatial types, the objects shown in Fig. 5 (a) can be represented correctly as shown in Fig. 5 (c).

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 *MBR*s can, which may also cause

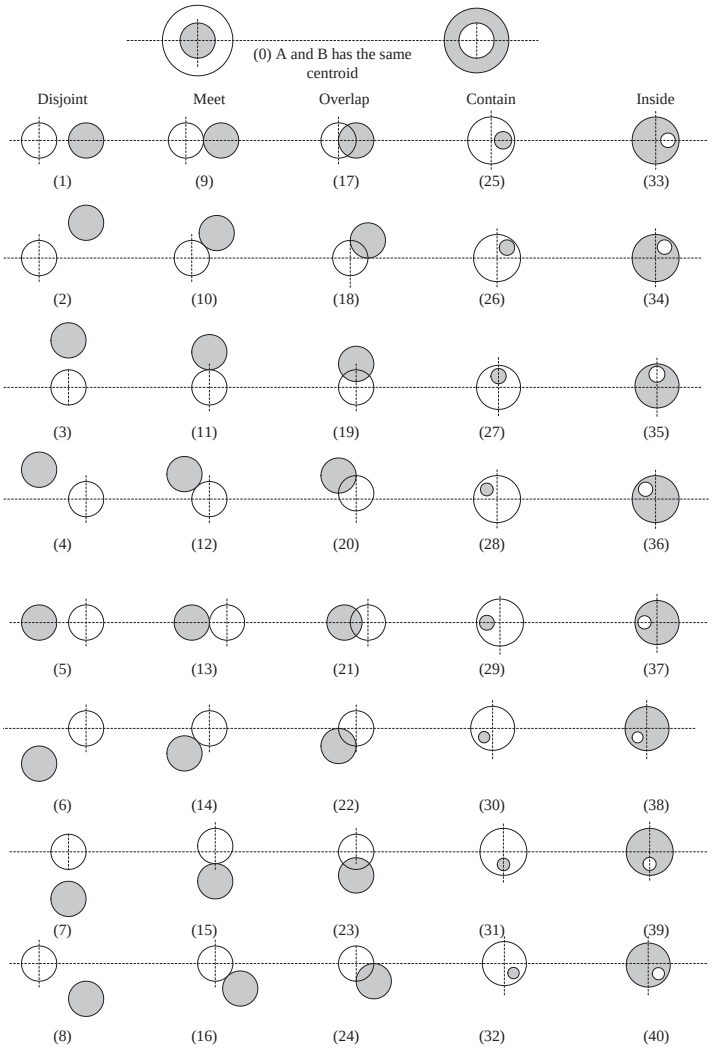
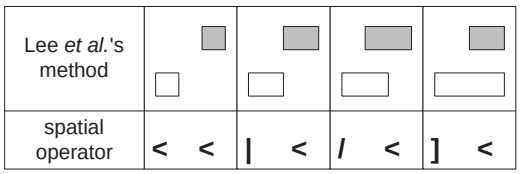
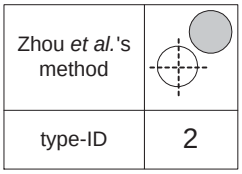


Fig. 6. 41 types of spatial relationships in 2D space.

ambiguity. For example, the spatial relationships between the white and gray objects shown in Fig. 7 (a) are different; however, these four different spatial relationships are classified as the same type (type-ID 2) by Zhou *et al.*'s method, as shown in Fig. 7 (b). The other disadvantage is that we find one missing spatial relationship with Zhou *et al.*'s method. Take Fig. 8 for example. Because the two objects shown in Fig. 8 (a) have the same centroid, the spatial relationship between them is type-ID 0, as shown in Fig. 8 (b). In fact, the spatial relationship between these two objects should be the one shown in Fig. 8 (c). However, this type of spatial relationship belonging to the partial overlap category does not exist in their proposed 41 spatial relationships.

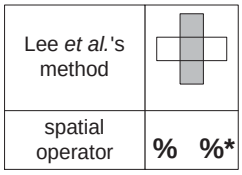


(a) 4 different spatial relationships encountered with Lee *et al.*'s method.

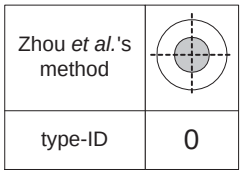


(b) One unique code (type-ID 2) with Zhou *et al.*'s method.

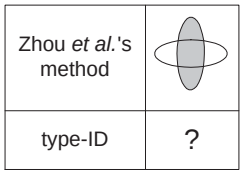
Fig. 7. An ambiguous case encountered with Zhou *et al.*'s method.



(a) The (%, %*) spatial relationship with Lee *et al.*'s method.



(b) The type-ID 0 spatial relationship with Zhou *et al.*'s method.



(c) An unknown type-ID (spatial relationship) with Zhou *et al.*'s method.

Fig. 8. The missing spatial relationship with Zhou *et al.*'s method.

2.2 Signatures

In [21], Lee *et al*. proposed the 2D B-string-based signature. Their approach uses superimposed coding and disjoint coding to speed up access. The integrated signature, as shown in Fig. 9, can handle retrieval by means of objects, binary spatial relationships and subpictures to distinguish different types of similarity.

object	type-0	type-1	type-2
RS^{obj}	RS^0	RS^1	RS^2
W_{obj} bits	W_0 bits	W_1 bits	W_2 bits

Fig. 9. The structure of the 2D B-string-based signature.

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

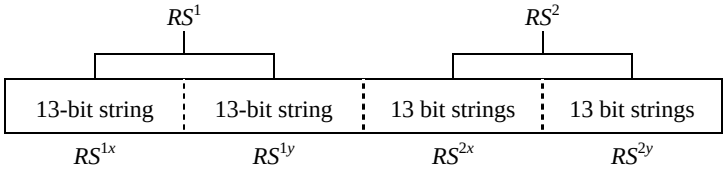


Fig. 10. The structure of the unique-ID-based signature.

3. THE PROPOSED METHOD

From the above observations, we are motivated to find a way to integrate the advantages of 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 of the 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 the 169 spatial relationships enclosed in bold boxes in Fig. 3 are difficult to classify.

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 the record signature is constructed. Finally, we present 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], a situation where *A* is both to the east of *B* and to the west of *B* (as shown in Fig. 11, for example) may occur. This situation, however, is not logical 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 the *x*-axis. (2) *A* is to the west of *B* iff: *centroid*(*A*) < *centroid*(*B*) on the *x*-axis. (3) *A* is to the north of *B* iff: *centroid*(*A*) > *centroid*(*B*) on the *y*-axis. (4) *A* is to the south of *B* iff: *centroid*(*A*) < *centroid*(*B*) on the *y*-axis.

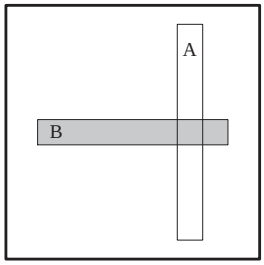


Fig. 11. An example of an orthogonal relationship.

From the above discussion, we can extend and revise the existing similarity of types 0, 1, and 2, which were introduced in [21], to obtain 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 ($T'_{AB} = T_{AB}$).

In 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 the 9 direction codes between the centroid of A and the centroid of B; (4) R_{AB} denotes one of the 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 Fig. 12, f_0, f_1', $f_{1.5}$, f_2', $f_{2.5}$, and f_3 are type-0, 1′, 1.5, 2′, 2.5, and 3 similarity, respectively. Fig. 13 shows the hierarchy of the definition of extended type-i similarity.

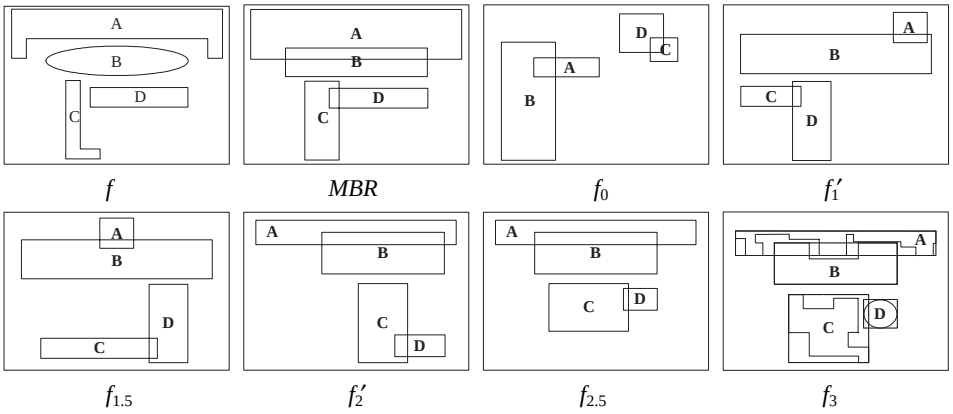


Fig. 12. Similarities.

There is a division at the *type*-1′ similarity. We can explain this situation by using Figs. 14 and 15. In Fig. 14, since object A in both pictures is to the northeast of object B, picture P exhibits type-1.5 similarity with picture Q. However, the spatial relationship between objects A and B in picture P is "$/^*$ $<^*$", while that in picture Q is "$|^*$ $<^*$".

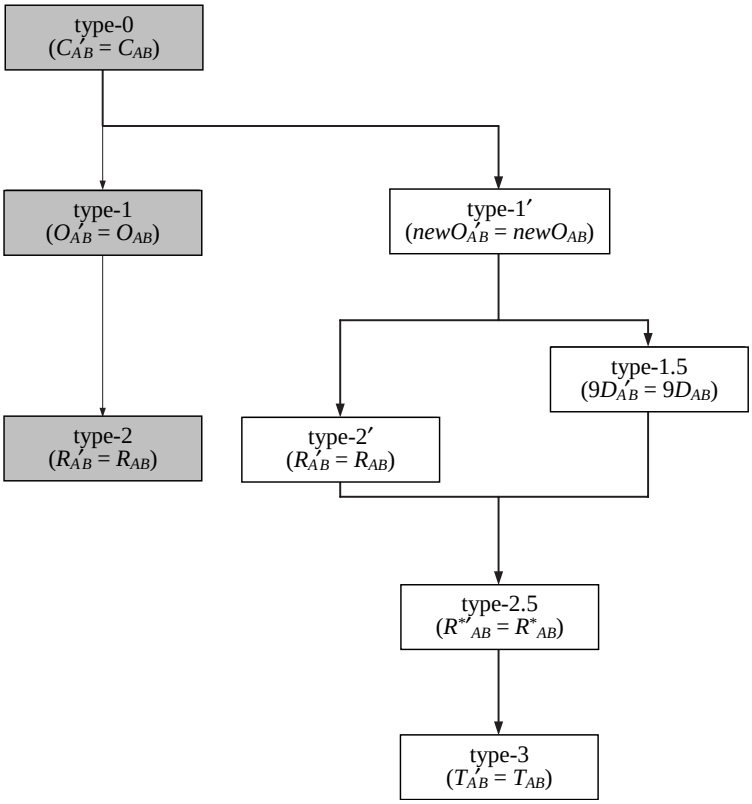
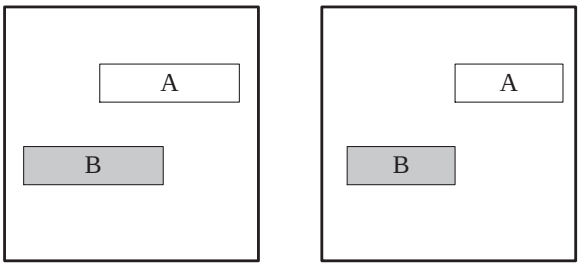


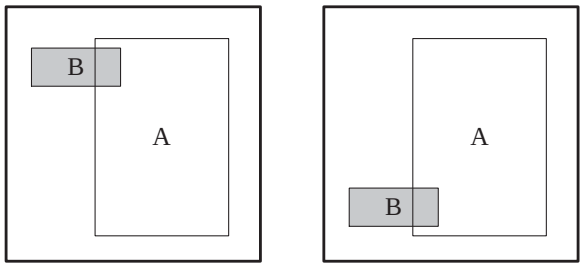
Fig. 13. The hierarchy of type-*i* similarity.



(a) Picture *P*. (b) Picture *Q*.

Fig. 14. An example of two pictures with type-1.5 similarity.

Therefore, picture *P* does not exhibit type-2′ similarity with picture *Q*. This indicates that pictures which exhibit type-1.5 similarity with each other may not have type-2′ similarity. In Fig. 15, object *B* in both pictures *P*′ and *Q*′ is to the west of object *A*. Moreover, the spatial relationship "/ %" between them in picture *P*′ is the same as that in picture *Q*′. Therefore, picture *P*′ has type-2′ similarity with picture *Q*′. However, object *B* in picture *P*′ is to the northwest of object *A*, while object *B* in picture *Q*′ is to the southwest of object *A*.



(a) Picture P'. (b) Picture Q'.

Fig. 15. An example of two pictures with type-2′ similarity.

Therefore, Picture P' does not exhibit type-1.5 similarity with picture Q', which indicates that some pictures which match with type-2′ similarity may not match with type-1.5 similarity. Note that although a set of pictures with type-1.5 similarity is not always a subset of pictures with type-2′ similarity and vice versa, we still put the position of type-2′ similarity lower than that of type-1.5 similarity in the hierarchical picture shown in Fig. 13. The reason is that pictures with type-1.5 similarity can only be distinguished based on 9 different spatial relationships, while pictures with type-2′ similarity can be classified based on 169 different spatial relationships.

The branches join together at type-2.5 similarity. The reason is that, by combining the 9 directional relationships (type-1.5) with the 169 spatial relationships in 2D space (type-2′), we can obtain up to 289 spatial relationships in 2D space, which can represent all types of spatial relationships more completely than the 169 spatial relationships can. The symbol R^* in Fig. 13 is used to represent one of the 289 spatial relationships. This relationship will be described later.

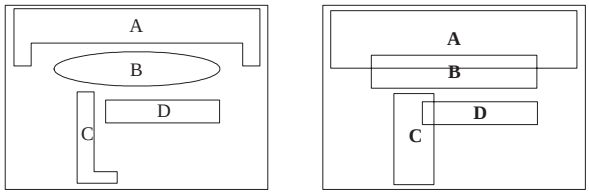
3.2 Four New Spatial Strings

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

3.2.1 Spatial category strings (*SCS*)

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

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



(a) The real picture. (b) The corresponding *MBR*.

Fig. 16. One example of picture *f*.

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

$$RSM = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 6 & 2 & 2 \\ 9 & 0 & 2 & 2 \\ 9 & 9 & 0 & 9 \\ 9 & 7 & 5 & 0 \end{bmatrix} \end{array}.$$

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

```
Function CATEGORY(Sid^x_{A,B}, Sid^y_{A,B})
1   if (Sid^x_{A,B} > 4) and (Sid^y_{A,B} > 4) then
2      if (7 ≤ Sid^x_{A,B} ≤ 10) and (7 ≤ Sid^y_{A,B} ≤ 10) then
3         return (2)    //* contain *//
4      else if (10 ≤ Sid^x_{A,B} ≤ 13) and (10 ≤ Sid^y_{A,B} ≤ 13) then
5            return (3)    //* belong *//
6         else return 4    //* partial overlap *//
7   else if (Sid^x_{A,B} > 2) and (Sid^y_{A,B} > 2) then
8         return (1)    //* join *//
9      else return (0)    //* disjoin *//
```

Fig. 17. The category function.

For the picture shown in Fig. 16, the corresponding spatial category matrix *C* is shown below, where 0 and 4 denote the join and partial overlap relationships, respectively:

$$C = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} - & - & - & - \\ 4 & - & - & - \\ 0 & 0 & - & - \\ 0 & 0 & 4 & - \end{bmatrix} \end{array}.$$

Based on the spatial category matrix, a *spatial category string SCS* is defined as $\{O_iO_jc_{ij} \mid c_{ij} \in \{0, 1, \ldots, 4\}\}$, where O_i and O_j are objects. Therefore, the corresponding *SCS* set for the spatial category matrix *C* is *SCS* = {*AB*4, *AC*0, *AD*0, *BC*0, *BD*0, *CD*4}.

3.2.2 Nine direction code strings (*DCS*)

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

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} - & - & - & - \\ 5 & - & - & - \\ 4 & 4 & - & - \\ 6 & 6 & 8 & - \end{bmatrix} \end{array}.$$

According to the 9DLT matrix *M*, a *nine direction code string DCS* is defined as $\{O_iO_jm_{ij} \mid m_{ij} \in \{0, 1, \ldots, 8\}\}$. Then, the corresponding *DCS* set for Fig. 16 is *DCS* = {*AB*5, *AC*4, *AD*6, *BC*4, *BD*6, *CD*8}.

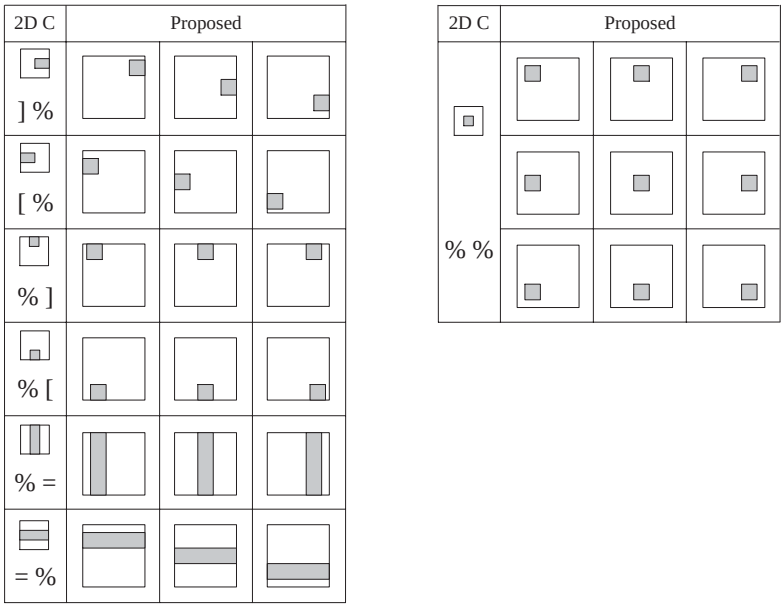
3.2.3 Identification number strings (*INS*)

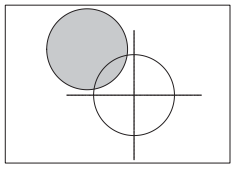
Based on the definition given in [20], there are 169 spatial relationships in 2D space. However, if we add the information about directional relationships, then some spatial relationships among these 169 spatial relationships will be divided into several more kinds of spatial relationships. Take the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

Therefore, the 169 spatial relationships adopted in the 2D C-string method [20] are not sufficient to represent spatial relationships in 2D space. In this way, we can preserve these operators used in the 169 spatial relationships; in addition, we can integrate the spatial category relationships and directional relationships to obtain one representation to divide spatial relationships in 2D space. Take Fig. 18 as an example. There are 17 spatial relationships which belong to the *partial overlap* spatial category, along with direction code 2 (i.e., *northwest*). Note that the number of spatial relationships in each division varies.

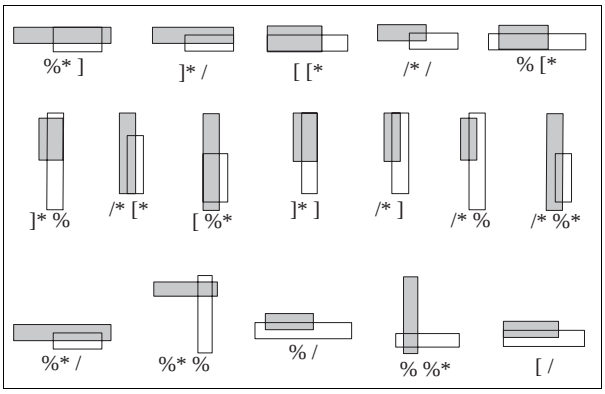
Therefore, since we take the 9 directions into consideration, the total number of spatial relationships in our proposed method is 289 (= 64 + 56 + 36 + 35 + 98), as summarized in Table 3. We carefully assign an identification number to each spatial relationship

Table 2. 27 cases in the *contain* category.





(a) Partial overlap category and direction code 2.



(b) 17 possible spatial relationships.

Fig. 18. One example of integrating a spatial category and directional relationships.

such that the same spatial operator with different direction codes is assigned the same identification number [28]. For example, the 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.

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	7	7	17	7	17

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	%* %	%* %	%* %	%* %	%* %	%* %	%* %	%* %	%* %

Because the information about spatial category relationships and the 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 *identification number string INS* as $\{O_iO_jid_{ij} \mid id_{ij} \in \{0, 1, \ldots, 16\}\}$. Consequently, the corresponding *INS* set for Fig. 16 is *INS* = {*AB*8, *AC*7, *AD*7, *BC*7, *BD*9, *CD*2}. Therefore, based on the specific combination of *SCS*, *DCS*, and *INS*, we can discriminate one of 289 spatial relationships between any two objects.

3.2.4 Topological relationship strings (*TRS*)

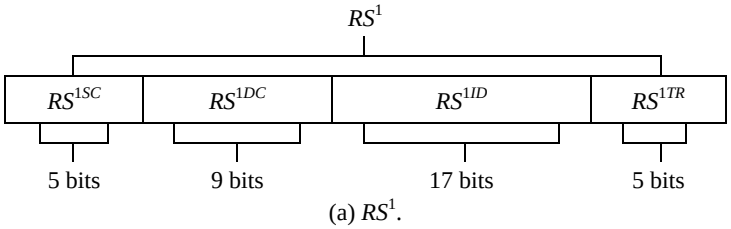
Basically, an $m \times m$ *Topological Relationship Matrix TRM* of picture *f* is similar to a *Spatial Category Matrix C*, except that the codes of the topology relationships are recorded instead of the codes of the spatial categories. For the picture shown in Fig. 16, the corresponding topological relationship matrix *TRM* is as follows:

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

Then, a *topological relationship string TRS* is defined as $\{O_iO_jt_{ij} \mid t_{ij} \in \{0, 1, \ldots, 4\}\}$. Therefore, the corresponding *TRS* set for the above topological relationship matrix *TRM* is *TRS* = {*AB*0, *AC*0, *AD*0, *BD*0, *BD*0, *BC*0}.

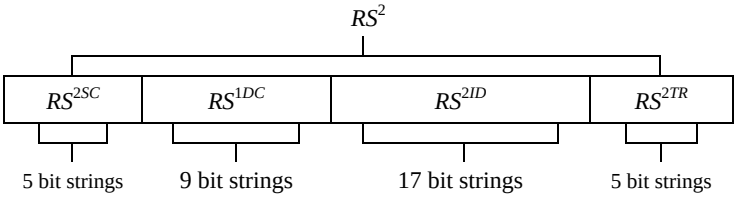
3.3 Record Signature

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



(a) RS^1.

Fig. 19. The structure of the proposed signature.



(b) RS^2.

Fig. 19. (Cont'd) The structure of the proposed signature.

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 the topological relationship
RS^{1SC}	the record signature flags from the viewpoint of the spatial category
RS^{1DC}	the record signature flags from the viewpoint of the 9 direction codes
RS^{1ID}	the record signature flags from the viewpoint of the 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_j^i	the RS^i for the jth picture
QRS	the query record signature

Algorithm Record Signature

Step 1: List all of *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 to a unique bit string.

Step 3: Set all bits in *RS* to 0.

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, we will use the following example. For the figure shown in Fig. 20, first, we construct the spatial matrix and the reduced spatial matrix (*RSM*):

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

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

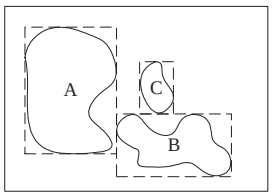


Fig. 20. An example

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* = {*AB*1, *AC*0, *BC*1}, *DCS* = {*AB*6, *AC*7, *BC*2}, *INS* = {*AB*1, *AC*10, *BC*0}, and *TRS* = {*AB*0, *AC*0, *BC*0}.
2. Design the function θ_r (where $b_r = 5$ and $k_r = 2$), 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 $ABi \in SCS$, then 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 $RS^2(i)$.
   RS = 11000 001000110 11000000001000000 10000
   10100 11101 01100 10001 10100 01100 10001 10100 11101.

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 consideration.

3.4 Object and Type-*i* Similarity Retrieval

We will use our proposed structure of a signature discussed in the previous section to illustrate the process of 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 into its *completed* form, instead of the *reduced* form. Given a record signature, we will now present an algorithm for converting such a record signature into the related object record signature.

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, we will use the following example. Suppose there are 8 pictures in the database, as shown in Fig. 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 $\theta_r(CD) = 00110$.

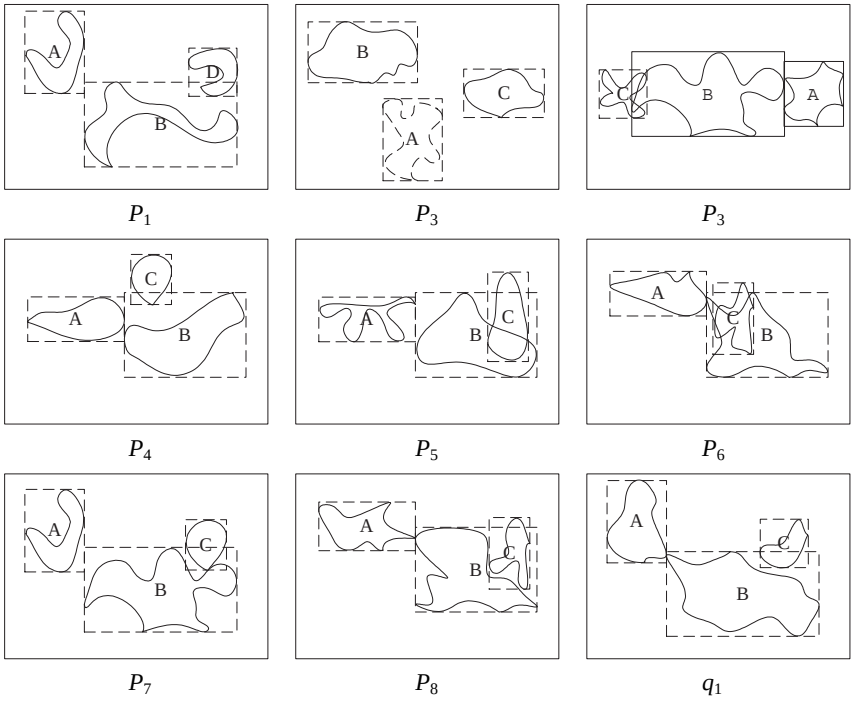


Fig. 21. An image database (P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8) and a query picture (q_1).

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

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

Next, we will show how the record signatures can be converted into 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 Fig. 21, the corresponding record signature is:

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

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 P_2 through P_8 may have the same objects as the query picture q_1, while picture P_1 may have some objects that are different from those in 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 Fig. 19.)

3.4.2 Query of type-0 similarity

The spatial category field (RS^{1SR} and RS^{2SR}) of the record signature contains information about the spatial category relationship between any two objects in the picture. Since the record signature has complete information about type-0 similarity (i.e., RS^{1SR} and RS^{2SR}), we only have to compare the related bit-strings to find possible answers. In the following, we will take the pictures and query shown in Fig. 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 need not to be checked further.
2. Next, we have $RS_1^{2SR} \cap QRS^{2SR} \neq QRS^{2SR}$ and $RS_3^{2SR} \cap QRS^{2SR} = QRS^{2SR}$, and the remaining pictures have the same result as picture P_3. We conclude that pictures P_3 through P_8 may have type-0 similarity with query picture q_1, while pictures P_1 and P_2 do not have type-0 similarity with query picture q_1.

3.4.3 Query of type-1′ similarity

Fig. 22 shows the structure of the type-1′ record signature *NSWE*. Given a record signature, we will now present an algorithm for converting such a record signature into the related type-1′ record signature.

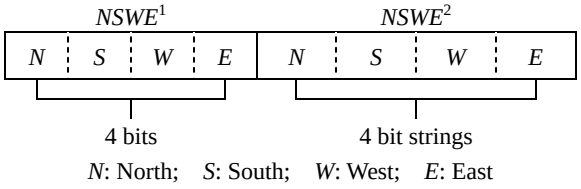


Fig. 22. Type-1′ signature *NSWE*.

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 $i \in \{1, 2, 8\}$ 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 $i \in \{4, 5, 6\}$ 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 $i \in \{2, 3, 4\}$ 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 $i \in \{6, 7, 8\}$ 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 the 8 pictures from P_1 through P_8, as shown in Fig. 21, only pictures P_3 through P_8 have type-0 similarity; therefore, we only have to check whether pictures P_3 through P_8 have type-1′ similarity with query picture q_1. We can now convert the record signatures of pictures P_3 through P_8 into the corresponding type-1′ signatures. Take picture P_3 as an example:

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

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

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

$NSWE_4$ = 1111 10101 10001 00101 10101,
$NSWE_5 = NSWE_7 = NSWE_8$ = 1101 00101 10101 00000 10101,
$NSWE_6$ = 1111 00101 10101 00101 10101.

Given query picture q_1 shown in Fig. 21, the corresponding type-1′ query signature is $QNSWE$ = 1101 00101 10101 00000 10101.

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

Therefore, according to the algorithm shown in Fig. 23, we can conclude that pictures P_4 through P_8 may have type-1′ similarity with query picture q_1. However, picture P_3 does not have type-1′ similarity with 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 all the information needed to find out if two objects have the same directional relationship. Taking advantage of this feature, we can answer a type-1.5 similarity retrieval query directly.

In Fig. 21, only pictures P_4 through P_8 have type-1′ similarity; therefore, we only need to find out which picture from P_4 through P_8 has type-1.5 similarity. The steps performed to check for 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 does not have type-1.5 similarity with query picture q_1. Pictures P_4, P_5, P_7, and P_8 need to be checked further.

```
01  If NSWE^1 ∩ QNSWE^1 = 0000 Then
02     Return ("the picture is not matched.");
03  For i := 0 to 3 do
04  Begin
05     j := (i + 1) mod 4; k := (j + 1) mod 4;
06     flag1 := 0; flag2 := 0; flag3 := 0;
07     If NSWE^1(i) ∩ QNSWE^1(i) = 1 Then
08        If NSWE^2(i) ∩ QNSWE^2(i) = $ Then flag1 := 1;
09     Else flag1 := 2;
10     If NSWE^1(j) ∩ QNSWE^1(j) = 1 Then
11        If NSWE^2(j) ∩ QNSWE^2(j) = $ Then flag2 := 1;
12     Else flag2 := 2;
13     If NSWE^1(k) ∩ QNSWE^1(k) = 1 Then
14        If NSWE^2(k) ∩ QNSWE^2(k) = $ Then flag3 := 1;
15     Else flag3 := 2;
16     flag := flag1 * flag2 * flag3;
17     If (flag ≠ 0 And flag ≠ 8) Then Return ("the picture is not matched.");
18  End;
19  Return ("two signatures have type-1′ similarity.");
```

[+] The symbol $ means that the intersection of two bit strings is not the union of some θ_r.

Fig. 23. Algorithm for type-1′ similarity retrieval.

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

3.4.5 Query of type-2′ similarity

Before checking for type-2′ similarity, we must check for type-1′ similarity. In subsection 3.4.4, we found that pictures P_4 through P_8 had type-1′ similarity with query picture q_1. To find out whether two objects have type-2′ similarity or not, we need information about the 169 spatial relationships between them. The identification number field of the record signature (RS^{1ID} and RS^{2ID}) provides with what we need. The steps performed to find out which picture has of type-2′ similarity with query picture q_1 are as follows.

1. Since $RS_i^{1ID} \cap QRS^{1ID} \neq QRS^{1ID}$, $i \in \{4, 5\}$, pictures P_4 and P_5 do not match and need not be checked further.
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 have type-2′ similarity with query picture q_1.

3.4.6 Query of type-2.5 similarity

Suppose the set $S_{1.5}$ contains the pictures which have type-1.5 similarity with query picture q_1 and that the set $S_{2'}$ contains the pictures which have type-2′ similarity with

query picture q_1. Thus, we have $S_{1.5} = \{P_5, P_7, P_8\}$ and $S_{2'} = \{P_6, P_7, P_8\}$. 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 have type-2.5 similarity with query picture q_1.

3.4.7 Query of type-3 similarity

Since pictures P_7 and P_8 have type-2.5 similarity with query picture q_1, pictures P_7 and P_8 are candidates for type-3 similarity checking. The topological relationship field of the record signature (RS^{1TR} and RS^{2TR}) can help us to perform type-3 similarity retrieval. 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 further.
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 found to have type-3 similarity with query picture q_1, but picture P_7 may not.

4. PERFORMANCE STUDY

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

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

In this experiments, we let the number of different kinds of objects appearing in the database be 60. For each object, the width and height were bounded between 1 and 100,000 units. We prepared 2,000 pictures using in our proposed method and Lee *et al.*'s method in the database in advance. We considered the case with 15 different objects randomly chosen, with uniform distribution in each picture. There were 100 query pictures, and each query picture contained 2 different objects. Thus, the maximum number of matching pictures was $100 \times 2{,}000 = 200{,}000$. The begin bound and end bound on the *x*- and *y*-coordinates of each object were randomly generated with uniform distribution. (Note that the data generated in our simulation study could be considered the result of a certain public database pre-processed using an image understanding technique which can identify and label objects [13]. The spatial relationship between any two objects can be derived from the coordinates of the objects.)

Table 6 compares the correct match rates obtained with these two methods. Each rate was calculated based on the fractional number shown below it. The denominator is the number of potentially matching pictures as determined by Lee *et al.*'s or our method. The numerator is the number of pictures that actually match the object or have 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 does. The numerator in our method for object similarity is the same as that in Lee *et al.*'s method, since both methods have the same definition of object similarity. The denominator in our method is smaller than that in Lee *et al.*'s method. Thus, our method has a higher correct

Table 6. A comparison of the correct match rates obtained with the 2D B-string-based method and our proposed signature method.

	object	type-0	type-1*	type-2*
2D B-based	12.10% (11,858/98,017)+	8.87% (5,403/60,902)	3.74% (465/12,430)	3.82% (298/7,810)
Proposed	80.86% (11,858/14,665)	40.37% (5,403/13,383)	30.27% (4,045/13,363)	5.41% (300/5,548)

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

Table 7. Correct match rates for type-1.5, type-2.5, and type-3 similarity retrieval obtained with on the proposed method.

type-1.5	type-2.5	type-3
12% (1,334/11,120)	4.54% (213/4,687)	3.33% (148/4,438)

match rate than Lee *et al.*'s method does for object similarity. Next, because we revised the definition of type-1 similarity used 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 smaller 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 by choosing a suitable size for a signature and for the hash function, which has been studied in [3].)

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

Next, we will discuss the storage cost of these two methods. With Lee *et al.*'s method, we let the length of the object signature be 15; we also let 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 of 2015 bits for a record signature for a picture. With our method, we let the length of a bit string be 55, resulting in a total of 2,016 bits (i.e., 36 bits for RS^1 and 36×55 bits for RS^2) for a record signature with a picture. Thus, the size of the signature in these two methods was almost the same.

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

Table 8. A comparison of the storage cost.

	Min.	Avg.	Max.
2D-B based	2,015	2,015	2,015
Proposed	1,411	1,523	1,631

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

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

In this experiment, there were 20 different objects and 2,000 pictures in the database. We considered the case in which each picture contains 5 different objects. There were 100 query pictures, and each query picture contained 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 was (13 + 13) + (13 + 13) * 100 = 2,626. We let the bit-string length for our method be 72. Then, the length of our signature was 2,628. (Note that in [10], the authors showed that the unique-ID-based signature method outperformed the 2D B-string-based signature method.) A comparison of the correct match rate obtained with the unique-ID-based method and our proposed method 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 achieves a higher correct match rate than the unique-ID-based method does. The correct match rate is affected by many parameters, so those for our method shown Tables 6 and 9 are different. As for the storage cost, with both methods, if $RS^1(i) = 0$, then $RS^2(i)$ can be removed.

Table 9. A comparison of correct match rates obtained with the unique-ID-based (UID-based) method and our proposed signature method.

	object	type-0	type-1*	type-2*
UID-based	51.76% (10,549/20,382)	42.51% (4,974/11,701)	23.28% (1,383/5,941)	20.98% (292/1,392)
Proposed	51.79% (10,549/20,370)	65.01% (4,974/7,651)	72.35% (3,774/5,216)	49.48% (1,229/1,491)

4.3 The Block Signature

The above approach to data filtering for all of the methods is based on a type of storage organization based on *sequential* signatures, as in *Quick Filter* [26]. That is, given *NR* records, we have to compare signatures *NR* times sequentially. To reduce the number of comparisons with each record signature in an image database, we can use

block signature [8, 10], multi-level signature [13], or dynamic hashing [26] techniques to each methods. Here, we will present simulation results obtained by applying the block signature (*BS*) technique in our method. The algorithm for constructing *BS* is almost the same as that for constructing *RS*. The only difference between them is that we use another hashing function, θ_b, according to the given k_b (the number of 1's in the block signature) and b_b (the number of bits in 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.

We considered three cases with 5, 10, and 15 objects in each picture, respectively. There were 2,000 pictures and 2,000 record signatures in the database. Each block signature recorded the information of 10 pictures. Thus, we had 2,000/10 = 200 block signatures and a total of 2,200 signatures in the database. From the simulation results shown in Table 10, we observe that the fewer the objects per picture, the smaller the 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.

Table 10. Percentages of signature comparisons for different numbers of objects per picture.

	object	type-0	type-1′	type-1.5	type-2′	type-2.5	type-3
5 objects	31.25%	24.58%	23.78%	12.25%	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%

Table 11. Percentages of signature comparisons for different numbers 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%	23.54%	21.30%	20.80%
15 records	77.46%	72.42%	72.36%	65.25%	41.03%	38.23%	37.10%

We also considered three cases with a block signature recording the information of 5, 10, and 20 records, respectively. Therefore, there were 400 (= 2,000/5), 200 (= 2,000/10), and 100 (= 2,000/20) block signatures in the database, respectively. The total number of signatures in the database for each case was 2,400, 2,200, and 2,100, respectively. There were 10 objects in each picture. From the simulation results shown in Table 11, we observe that the more records recorded in one block signature, the larger the 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, a given number of records recorded in one block signature, denoted by *NRperB*, and a given rate of matched 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 an additional effect on *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 1,000 records, 10 objects and a block signature containing the information of every 10 records, we can 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, 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 that applies block signatures always requires a smaller number of comparisons than the sequential signature method does.

Table 12. Match rates 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%

Table 13. Block signatures match rates for different numbers 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%
15 records	76.33%	71.04%	70.98%	63.15%	38.08%	35.14%	33.96%

5. CONCLUSIONS

In this paper, we have presented a new method which combines the advantages of the previous 2D C-string, 9DLT matrix, and DT methods for similarity retrieval from a large image database. We have extended the existing three types of type-*i* similarity to obtain up to six types to facilitate highly accurate similarity retrieval. After adding 9 directions to 169 spatial relationships, we have found that up to 289 spatial relationships can be used to represent the spatial relationships in 2D space. This makes it possible to

distinguish some spatial relationships that can not be distinguished based on the 169 spatial relationships adopted in the 2D C-string method. Moreover, in order to overcome the ambiguity resulting from enclosing symbolic objects with *MBR*s, we have adopted the concept of topological relationships. Based on the above extensions, we have proposed a new signature structure and algorithms for obtaining object and six type-*i* similarities. From our simulation results, we have shown that our proposed method achieves higher correct match rates than the 2D B-string-based and unique-ID-based signature methods do. Handling similarity retrieval in the case of images which may be rotated will be a subject of 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*, Vol. 17, 1999, pp. 967-980.
2. I. Ahmad and W. I. Grosky, "Indexing and retrieval of images by spatial constraints," *Journal of Visual Communication and Image Representation*, Vol. 14, 2003, pp. 291-320.
3. C. C. Chang and H. C. Wu, "A module-oriented signature extraction to retrieve symbolic pictures," *Journal of Computer*, Vol. 2, 1990, pp. 45-54.
4. C. C. Chang, "Spatial match retrieval of symbolic pictures," *Journal of Information Science and Engineering*, Vol. 7, 1991, pp. 405-422.
5. C. C. Chang and C. F. Lee, "A spatial match retrieval mechanism for symbolic pictures," *The Journal of Systems and Software*, Vol. 44, 1998, pp. 73-83.
6. S. K. Chang, Q. Y. Shi, and C. W. Yan, "Iconic indexing by 2D strings," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, 1987, pp. 413-428.
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, 1997, pp. 95-105.
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, 2000, pp. 1263-1276.
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, 2002, pp. 117-147.
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, 2004, pp. 1633-1645.
12. M. J. Egenhofer, "Point-set topological spatial relations," *International Journal of Geographical Information Systems*, Vol. 5, 1991, pp. 161-174.
13. A. El-Kwae and M. R. Kabuka, "Efficient content-based indexing of large image databases," *ACM Transaction on Information Systems*, Vol. 18, 2000, pp. 171-210.
14. C. Faloutsos and S. Christodoulakis, "Description and performance analysis of sig-

nature file methods for office filing," *ACM Transactions on Office Information Systems*, Vol. 5, 1987, pp. 237-257.
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 Computer Maganize*, Vol. 28, 1995, pp. 23-32.
16. S. Guru and P. Nagabhushan, "Triangular spatial relationship: a new approach for spatial knowledge representation," *Pattern Recognition Letters*, Vol. 22, 2001, pp. 999-1006.
17. P. W. Huang and C. H. Lee, "Image database design based on 9D-SPA representation for spatial relations," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, 2004, pp. 1486-1496.
18. J. S. Jin and R. Kurniawati, "Varying similarity metrics in visual information retrieval," *Pattern Recognition Letters*, Vol. 22, 2001, pp. 583-592.
19. J. T. Lee and H. P. Chiu, "2D Z-string: a new spatial knowledge representation for image databases," *Pattern Recognition Letters*, Vol. 24, 2003, pp. 3015-3026.
20. S. Y. Lee and F. J. Hsu, "2D C–string: a new spatial knowledge representation for image database systems," *Pattern Recognition*, Vol. 23, 1990, pp. 1077-1087.
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, 1992, pp. 373-397.
22. S. Y. Lee, M. C. Yang, and J. W. Chen, "2D B-string: a spatial knowledge representation for image database systems," in *Proceeding of 2nd International Computer Science Conference* (*ICSC '92*), 1992, pp. 609-615.
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, 2001, pp. 153-163.
24. P. Maresca, A. Guercio, T. Arndt, and G. Tortora, "Multimedia indexing with the SMART system," *Journal of Visual Languages and Computing*, Vol. 11, 2000, pp. 405-438.
25. G. M. Petrakis, "Design and evaluation of spatial similarity approaches for image retrieval," *Image and Vision Computing*, Vol. 20, 2002, pp. 59-76.
26. F. Rabitti and P. Zezula, "A dynamic signature technique for multimedia databases," in *Proceeding of 13th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, 1989, pp. 193-210.
27. Y. H. Wang, "Image indexing and similarity retrieval based on spatial relationship model," *Information Sciences*, Vol. 154, 2003, pp. 39-58.
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, 2001.
29. A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 11, 1999, pp. 81-93.
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, 1997, pp. 751-758.
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, 2001, pp. 469-477.

Wei-Horng Yeh (葉韋宏) was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. degree in Applied Mathematics in 1999, and the M.S. degree in Computer Science and Engineering in 2001, all from National Sun Yat-Sen University in Kaohsiung, Taiwan, R.O.C. He is currently a Ph.D. candidate in the Department of Computer Science and Engineering, National Sun Yat-Sen University. His research interest includes the index design for image and video databases.

Ye-In Chang (張玉盈) was born in Taipei, Taiwan, R.O.C., in 1964. She received the B.S. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1986, and M.S. and Ph.D. degrees in Computer and Information Science from the Ohio State University, Columbus, OH, in 1987 and 1991, respectively. From August 1991 to July 1999, she joined the faculty of the Department of Applied Mathematics at National Sun Yat-Sen University, Kaohsiung, Taiwan. Since August 1997, she has been a Professor in the Department of Applied Mathematics at National Sun Yat-Sen University, Kaohsiung, Taiwan. Since August 1999, she has been a Professor in the Department of Computer Science and Engineering at National Sun Yat-Sen University, Kaohsiung, Taiwan. Her research interests include database systems, distributed systems, multimedia information systems, and mobile information systems.