Spatial-match Iconic Image Retrieval with Ranking in Multimedia Databases

Jae-Woo Chang* and Yeon-Jung Kim
Department of Computer Engineering, Chonbuk National University
Chonju, Chonbuk 560-756, South Korea
e-mail : jwchang@dblab.chonbuk.ac.kr
Phone : +82-652-270-2414
FAX : +82-652-270-2418

Abstract

In multimedia database applications, it is necessary to retrieve multimedia documents to satisfy a user's query based on image content. For this, we represent an image as a set of meaningful graphic descriptions, called icon objects, and then do indexing by regarding the icon object as a representative of a given document. When users request content-based image retrieval, we convert a query image into a set of icon objects and retrieve relevant images in the multimedia database. In this paper, we propose new spatial-match iconic image representation schemes supporting ranking, called SRC scheme using MBC(Minimum Bounding Circle) and SRR scheme using MBR(Minimum Bounding Rectangle). Because our SRC and SRR schemes combine directional operators with positional operators, they can represent spatial relationships between icon objects precisely and can provide ranking for the retrieved images. In addition, we compare our schemes with the conventional 9DLT and SMR schemes in terms of retrieval effectiveness. Finally, we show from our experiment that our schemes hold about 7-25% higher recall and about 10-20% higher precision than the 9DLT and the SMR.

Keywords : spatial-match representation, iconic image retrieval, multimedia databases
General topic : Multimedia Databases, Multimedia Information Systems

*Correspondence to this author

1. Introduction

Recently, multimedia database systems have become a critical research area of computer systems. The applications of multimedia databases include digital library, advertisements, architectural design, cartography, video on demand (VOD), digital newspaper, cyber museum, and electronic commerce. Records (i.e., Documents) used in the multimedia database applications are typically complex in their structure and contain various multimedia data, such as text, image, audio, video, and animation. When the traditional database systems deal with multimedia documents, they can support only query by text content (e.g., captions and keywords) because they mainly focus on formatted data and texts. However, the approach using text content has a couple of problems [1]. First, original keywords do not allow for unanticipated searching. Secondly, the caption is not adequate to describe the layout, sketch, and shape of images. For the multimedia database applications, we can consider a query like this: *Retrieve images which contain trees on the north side of a house and cars on the left side of the house.* To support the query, it is essential to support content-based image retrieval on multimedia documents.

Given a pixel-level original image, a variety of image processing and understanding techniques are used to identify salient objects and their positions in the image. Though this task is computationally expensive and difficult, it is performed only at the time of image insertion into the multimedia database. Moreover, this task may be carried out in a semi-automated way or in a automated way, depending on the domain and complexity of the images. An iconic image is obtained by associating each salient object with a meaningful graphic description, called icon object. So, an iconic image representation can provide users with a high-level of image abstraction. The iconic image representation has a couple of advantages. First, the use of iconic images avoids the need for the repeated image understanding tasks. Processing an original image for interactive responses to high-level user queries is inefficient because the number of images tends to be large in most multimedia database applications. Secondly, the size of the iconic image being indexed is much smaller than that of the original image. Thus, the iconic image representation can be well suited to distributed database environments where we require a large number of image transmissions between distant nodes.

In general, all pixel-level original images are analyzed prior to storage so that icon objects can be extracted from their content and stored together with their original images. The icon objects are used to search the multimedia database and to determine whether an image satisfies a query's selection criteria. Ultimately, the effectiveness of multimedia database systems depends on the type and correctness of image content representation, the type of queries allowed, and efficiency of search techniques designed. The purpose of our paper is to provide both effective representation and efficient retrieval of images when an original image is automatically or manually transformed into its iconic image including only icon objects. For this, we propose new spatial-match iconic image representation schemes supporting ranking in multimedia databases. Because our spatial-match iconic image representation schemes combine directional operators with topological operators, they can represent spatial relationships between icon objects precisely. In addition, because our schemes support ranking for the retrieved images, we can provide the image results retrieved in the order of relevance to a user query. In order to accelerate image searching and support image ranking, we also design an efficient retrieval method based on an inverted file technique.

The remainder of this paper is organized as follows. In Section 2, we introduce a review of related work done in the area of iconic image representation, such as 9DLT and SMR. In Section 3, we propose new spatial-match iconic image representations with ranking in order to support content-based image retrieval in an effective way. In Section 4, we design an efficient retrieval method being able to make image searching fast and present algorithms for inserting and retrieving spatial strings into/from it. In Section 5, we compare the performance of our schemes with those of the 9DLT and the SMR in term of both retrieval effectiveness and system efficiency. Finally, we draw our conclusions and suggest future work in Section 6.

2. Related Work

There are many researches on spatial-match iconic image representation schemes [2,3,4,5,6,7]. They are classified into mainly two approaches. The first approach belongs to one based on directional information, for instance, 9DLT(Direction Lower Triangular) scheme [4]. The other approach is based on topological information, for instance, 2D C-string [3] and SMR(Spatial Match Representation) schemes [5]. In this section, we introduce the 9DLT and the SMR, which give much impression to our work.

2.1 9DLT scheme

Chang and Jiang [4] proposed the 9DLT scheme to describe spatial relationships between objects, i.e., icon objects. In the scheme, direction codes are denoted by nine integers, i.e., 1, 2, 3, 4, 5, 6, 7, 8, and 0, according to the direction of a target object from the reference object. Figure 1 shows the nine direction codes.

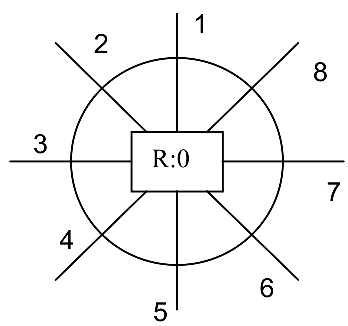


Figure 1. 9DLT directional codes

Here, R is the reference object. For instance, the direction code 1 indicates that the target object is located at the north side from R, and the direction code 2 indicates the north-west side from R. The direction code 0 indicates that the target object and R are located at the same position in an iconic image. Using the directional codes, the 9DLT scheme easily describes spatial relationships between objects, being represented by (*object A, object B,* D_{AB}). Here, D_{AB} means the *9DLT directional code*, being one of integers from 0 to 8, which indicates spatial relationships between objects in terms of direction. The formal notations of the spatial relationships between object A and object B are as follows :

(type-0) $ST_0^{AB} = \{(A, B, D'_{AB})\}$, where

$D'_{AB} = 0$ if $D_{AB} = 0$

$D'_{AB} = 0, 1$ if $D_{AB} = 1$

$D'_{AB} = 0, 3$ if $D_{AB} = 3$

$D'_{AB} = 0, 5$ if $D_{AB} = 5$

$D'_{AB} = 0, 7$ if $D_{AB} = 7$

$D'_{AB} = 0, 1, 2, 3$ if $D_{AB} = 2$

$D'_{AB} = 0, 3, 4, 5$ if $D_{AB} = 4$

$D'_{AB} = 0, 5, 6, 7$ if $D_{AB} = 6$

$D'_{AB} = 0, 1, 7, 8$ if $D_{AB} = 8$

(type-1) $ST_1^{AB} = \{(A, B, D_{AB})\}$

(type-2) $ST_2^{AB} = \{(A, B, D_{AB}, SC_X^{AB}, SC_Y^{AB})\}$, where

$SC_X^{AB} = 0$ if $|r_X(A) - r_X(B)| <= 1$,

$SC_X^{AB} = 1$ if $|r_X(A) - r_X(B)| > 1$,

$SC_Y^{AB} = 0$ if $|r_Y(A) - r_Y(B)| <= 1$,

$SC_Y^{AB} = 1$ if $|r_Y(A) - r_Y(B)| > 1$

Here, ST_i^{AB} represents the *type-i* spatial strings for objects A and B and SC_X^{AB} and SC_Y^{AB} represent the spatial codes for objects A and B in the X-axis and the Y-axis, respectively. Expression $|t|$ denotes the absolute values of *t;* for example, $|-2| = 2$. The type-0, 1, and 2 spatial strings can be used for approximate match, exact match, and distance-considering exact match, respectively.

2.2 SMR scheme

Because the 9DLT scheme represented each salient object of an original image as its icon object being expressed as a point, it can not express spatial relationships between icon objects by considering the size of salient objects in the original image. The SMR scheme proposed by Chang et al. [5] makes use of fifteen topological operators to describe spatial relationships between icon objects. The topological operators are made by applying the operators used for the specification of temporal relationships between time intervals to spatial environments [8]. Figure 2 shows the fifteen topological operators for the SMR scheme. A topological operator P_X (P_Y)denotes relationships between the projections *p* and *q*, respectively, of object A and B over the X-axis (Y-axis). The SMR scheme represents spatial string for both exact match and approximate match as follows:

- Exact match string :
  $ST_E^{AB} = \{(Object\ A, Object\ B, P_X, P_Y)\}$
- Approximate match string :
  $ST_A^{AB} = \{(Object\ A, Object\ B, PA_X, PA_Y)\}$

Here $PA_X(PA_Y)$ is a topological operator for approximate-match as shown in Table. For instance, if an objects A is located far away from an object B over the X-axis as well as the Y-axis, the exact match string for spatial relationship between the objects A and B can be represented as (*A B 0 0*), where A and B indicate objects, the third term 0 indicates P_X, and the fourth term 0 indicates P_Y. If the exact match string is (*A B 0 0*), the approximate match strings are (*A B 0 0*), (*A B 1 0*), and (*A B 0 1*).

Table 1. Exact-match and approximate-match operators in the SMR scheme.

Operators	P_X (P_Y)	PA_X(PA_Y)	Operators	P_X (P_Y)	PA_X(PA_Y)
p>>q	0	0, 1	p>+q	1	0, 1, 2
p>=q	2	1, 2, 3	p>q	3	2, 3, 4, 5
p>-q	4	3, 4, 7, 8	p>|q	5	3, 5, 6, 7
p><q	6	5, 6, 9	p=q	7	4, 5, 7, 9, 10
p<>q	8	4, 8, 10	p<|q	9	6, 7, 9, 11
p<-q	10	7, 8, 10, 11	p<q	11	9, 10, 11, 12
p<=q	12	11, 12, 13	p<+q	13	12, 13, 14
p<<q	14	13, 14			

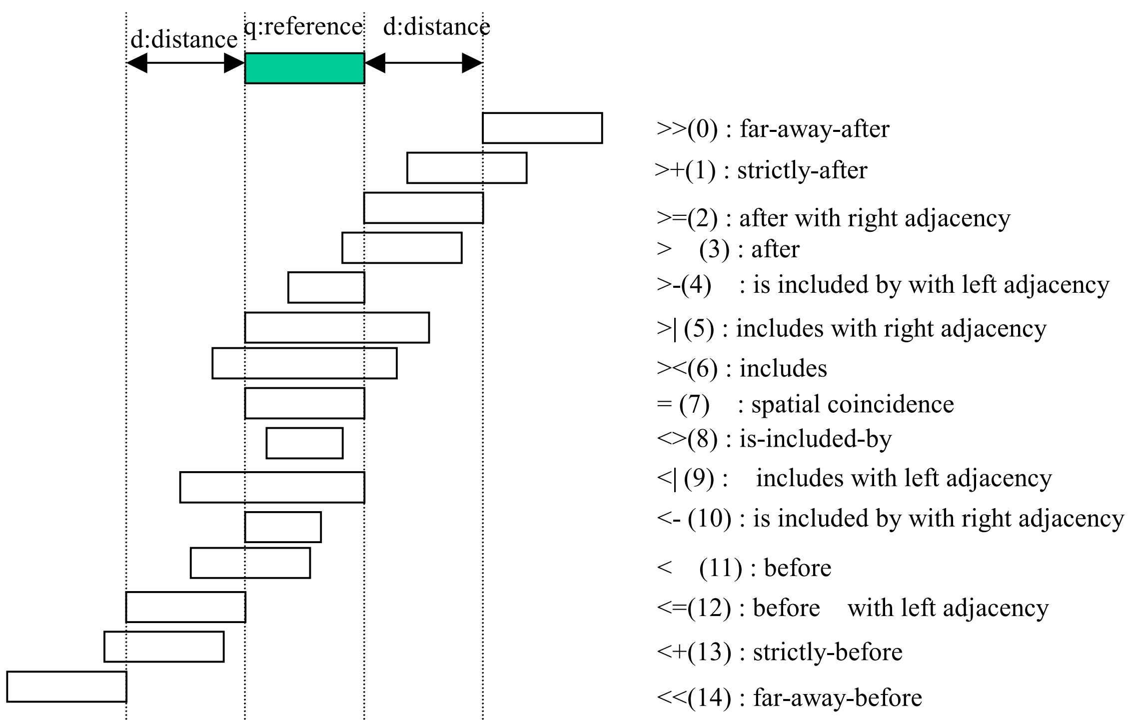


Figure 2. Positional operators in the SMR scheme

3. New Spatial-match Iconic Image Representations with Ranking

For image indexing, a large number of known image processing and understanding techniques can first be used to identify some salient objects and their relationships in an original image. Next, we can easily obtain an iconic image by associating a meaningful icon object with each salient object in the original image. By using some spatial-match representation schemes, we can finally obtain spatial strings from spatial relationships between icon objects. For image retrieval, a user query can first be transformed into an iconic query image in the same way as that used in the image indexing. Next, we can represent the iconic query image as spatial strings by using some spatial-match representation schemes. For this, the 9DLT and the SMR have a critical problem that they can not support ranking on the retrieved image results since they adopt a signature file as their access method. In this section, we propose new spatial-match iconic image representation schemes which represent spatial relationships between icon objects by using both directional and positional operators. Our representation schemes can also support ranking for the retrieved results to a user query by adopting an inverted file as their access method. We call our representation schemes SRC(**S**patial-match Representation Supporting **R**anking with Minimum Bounding **C**ircle) and SRR(**S**patial-match Representation Supporting **R**anking with Minimum Bounding **R**ectangle), respectively.

3.1 Representation of spatial relations in the SRC scheme

In order to identify an object in an image, we use Minimum Bounding Circle(MBC)[9], being a circle which surrounds the object. Using MBC, we can represent the spatial relationships between two objects based on the projections of the objects on the X-axis and the Y-axis. In Figure 3, rc_X and rc_Y are the projected center point of the reference object on the X-axis and the Y-axis, respectively. Similarly, tc_X and tc_Y are the projected center point of the target object. rad_R is the radius value of the reference object and rad_T is the radius value of the target object. *dis* is the distance between the center points of the reference object and that of the target object, i.e., $dis = \sqrt{dis_X^2 + dis_Y^2}$. In addition, *Ang* is measured as an angle made between the centers of two objects when the reference object is considered as the center of the XY-coordinates.

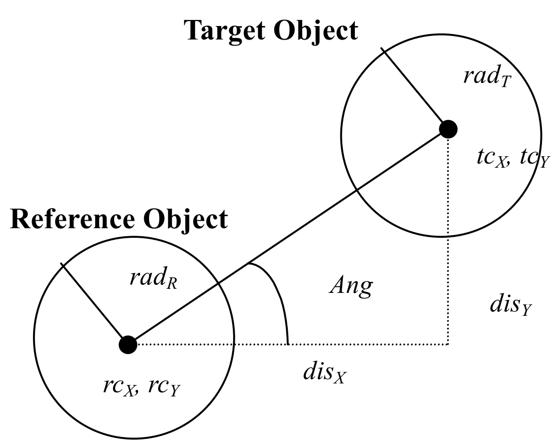


Figure 3. Spatial relation between two objects in the SRC scheme

For spatial-match representation in the SRC scheme, we define seven positional operators such as *FA*(*Far-away*), *DJ*(*Disjoint*), *ME*(*Meet*), *OL*(*Overlap*), *CL*(*Is-included-by*), *IN*(*Include*), *SA*(*Same*). They are made from the positional operators of the SMR scheme in order to represent spatial relationship between two objects in the XY-coordinates. The spatial relation R^{AB} between two objects is determined as follows. Here DM means a distance multiplier for deciding *far-away*.

- $R^{AB} = FA$ iff $dis \geq (rad_R + rad_T) * DM$
- $R^{AB} = DJ$ iff $(rad_R + rad_T) < dis \leq (rad_R + rad_T) * DM$
- $R^{AB} =$ ME iff $dis = (rad_R + rad_T)$
- $R^{AB} = OL$ iff $(dis<rad_R+rad_T)\wedge(((rad_R\leq rad_T)\wedge(rad_R+dis>rad_T))\vee((rad_R>rad_T)\wedge(rad_T+dis>rad_R)))$
- $R^{AB} =$ IN iff $(rad_R > rad_T)\wedge(rad_T + dis \leq rad_R)$
- $R^{AB} =$ CL iff $(rad_T > rad_R)\wedge(rad_R + dis \leq rad_T)$
- $R^{AB} = SA$ iff $(rad_T = rad_R)\wedge(dis = 0)$

3.2 Representation of spatial relations in the SRR scheme

In order to identify a salient object in an image, a technique is needed to put the object in Minimum Bounding Rectangle (MBR)[10]. In other words, MBR is the rectangle that surrounds the object in terms of both the lower left corner (the minimum point on the X-axis and the Y-axis) and the upper right corner (the maximum point on the X-axis and the Y-axis). Using MBR, we can represent the spatial relationships between two objects based on the projections of the objects on the X-axis and the Y-axis. In Figure 4, rs_i and re_i are the projected minimum and maximum values of the reference object on the i-axis (i= X or Y), respectively. ts_i and te_i are the minimum value and the maximum value of the target object on the i-axis, respectively. As a result, rs_X is the projected minimum value of the reference object on the X-axis and re_X is the projected maximum value of the reference object on the X-axis. The same meaning is applied to rs_Y, re_Y, ts_X, te_X, ts_Y, te_Y.

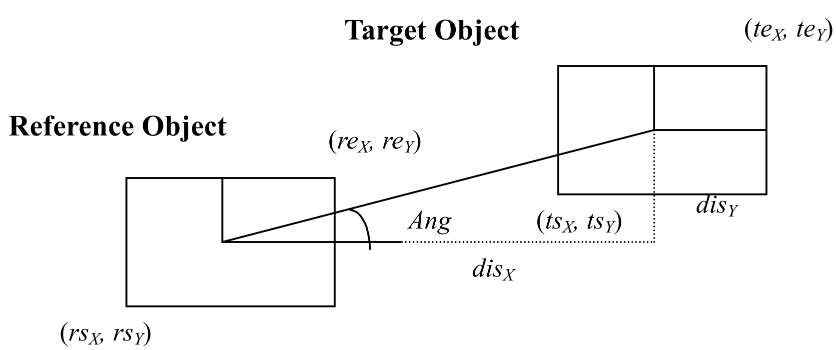


Figure 4. Spatial relation between two objects in the SRR scheme

Also, dis_X means the distance between the centers of two objects on the X-axis and dis_Y on the Y-axis. In addition, *Ang* is an angle measured between the centers of two objects when the center of reference object is considered as the center of the XY-coordinates. Based on the SMR positional operators [3], we use seven positional operators such as *FA*(*Far-away*), *DJ*(*disjoint*), *ME*(*Meet*), *OL*(*Overlap*), *CL*(*Is Included by*), *IN*(*Include*), *SA*(*Same*). The spatial relation R_i^{AB} between two objects for the i-axis is determined as follows:

- $R_i^{AB} = FA$ iff $dis_i \geq (re_i - rs_i + te_i - ts_i)/2*DM$

- $R_i^{AB} = DJ$ iff $(re_i - rs_i + te_i - ts_i)/2 < dis_i < (re_i - rs_i + te_i - ts_i)/2*DM$
- $R_i^{AB} = ME$ iff $dis_i = (re_i - rs_i + te_i - ts_I)/2$
- $R_i^{AB} = OL$ iff $(rs_i \leq ts_i \wedge re_i < te_i) \vee (rs_i < ts_i \wedge re_i \leq te_i) \vee (rs_i \geq ts_i \wedge re_i > te_i) \vee (rs_i > ts_i \wedge re_i \geq te_i)$
- $R_i^{AB} = CL$ iff $(rs_i \geq ts_i \wedge re_i < te_i) \vee (rs_i > ts_i \wedge re_i \leq te_i)$
- $R_i^{AB} = IN$ iff $(rs_i \leq ts_i \wedge re_i > te_i) \vee (rs_i < ts_i \wedge re_i \geq te_i)$
- $R_i^{AB} = SA$ iff $(dis_i = 0) \wedge (re_i - rs_i = te_i - ts_i)$

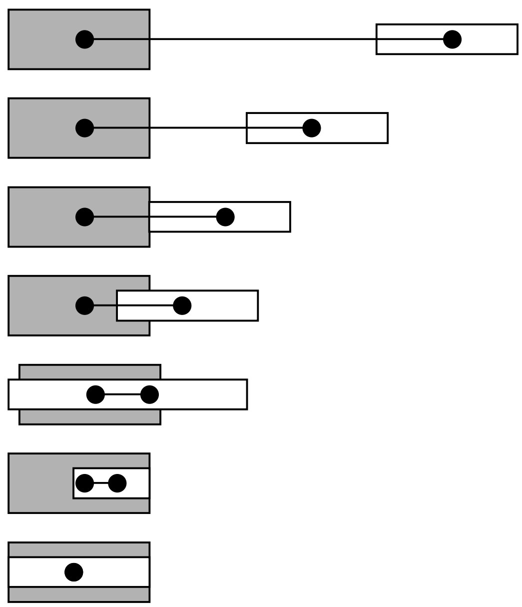


Figure 5. Positional operators in our schemes

Figure 5 depicts the seven positional operators with which we can determine two-dimensional spatial relations on the XY-coordinates. The two-dimensional spatial relation R^{AB} between two objects is defined as follows. Here, R_X^{AB} and R_Y^{AB} indicate the spatial relationships between two objects A and B on the X-axis and the Y-axis, respectively.

- $R^{AB} = FA$ iff $R_X^{AB} = FA \vee R_Y^{AB} = FA$
- $R^{AB} = DJ$ iff $(R_X^{AB} \neq FA \wedge R_Y^{AB} = DJ) \vee (R_Y^{AB} \neq FA \wedge R_X^{AB} = DJ)$
- $R^{AB} = ME$ iff $(R_X^{AB} \neq FA \wedge R_X^{AB} \neq DJ \wedge R_Y^{AB} = ME) \vee (R_Y^{AB} \neq FA \wedge R_Y^{AB} \neq DJ \wedge R_X^{AB} = ME)$
- $R^{AB} = OL$ iff $(R_X^{AB} \neq FA \wedge R_X^{AB} \neq DJ \wedge R_X^{AB} \neq ME \wedge R_Y^{AB} = OL)$
  $\vee (R_Y^{AB} \neq FA \wedge R_X^{AB} \neq DJ \wedge R_Y^{AB} \neq ME \wedge R_X^{AB} = OL)$
  $\vee (R_Y^{AB} \neq CL \wedge R_X^{AB} = IN) \vee (R_Y^{AB} \neq IN \wedge R_X^{AB} = CL)$
- $R^{AB} = CL$ iff $(R_X^{AB} = CL \wedge R_Y^{AB} = SA) \vee (R_X^{AB} = SA \wedge R_Y^{AB} = CL) \vee (R_X^{AB} = CL \wedge R_Y^{AB} = CL)$
- $R^{AB} = IN$ iff $(R_X^{AB} = IN \wedge R_Y^{AB} = SA) \vee (R_X^{AB} = SA \wedge R_Y^{AB} = IN) \vee (R_X^{AB} = IN \wedge R_Y^{AB} = IN)$
- $R^{AB} = SA$ iff $R_X^{AB} = SA \wedge R_Y^{AB} = SA$

We finally define a spatial string of two objects A and B as follows. Here, *Ang* is ranged from 0^{o} to 360^{o}.

- *SS* = (Reference *Object A, Target Object B,* R^{AB}*, Ang*)

3.3 Similar topological operators

When we analyze the seven topological operators, some topological operators, i.e., *IN* and *SA,* may have their similarity on spatial relationship, but they are determined to be different positional operators. For this, we classify the seven topological operators into two groups, i.e., '*disjoint*' and '*include*'. The '*disjoint*' group includes *far-away(FA), disjoint(DJ), meet(ME),* and *overlap(OL)* operators. The '*include*' group includes *include(IN), is-included-by(CL)* and *same(SA)* operators. We can compute their similarity between two operators as follows.

1. In the '*disjoint*' group, we determine their similarity between the four topological operators. The topological operators can be ordered according to their similarity, i.e., *FA – DJ – ME – OL.* For instance, *FA* has the highest similarity with *DJ* while it has the lowest similarity with OL.
2. In the '*include*' group, we also determine their similarity between the three positional operators, in the order of *CL – SA – IN*. For instance, *CL* has the highest similarity with *SA* while having the lowest similarity with IN.
3. Finally, we determine that there is no similarity between one of the topological operators in the '*disjoint*' group and one in the '*include*' group, except *OL*. In case of *OL*, we determine that *OL* has its similarity to *CL and IN* of the '*include*' group.

Figure 6 depicts a graph to describe their similarity between seven topological operators based on the above procedure. Using Figure 6, we can compute the similarity distance (sim_dis) between positional operators as the number of edges between nodes, as shown in Table 2.

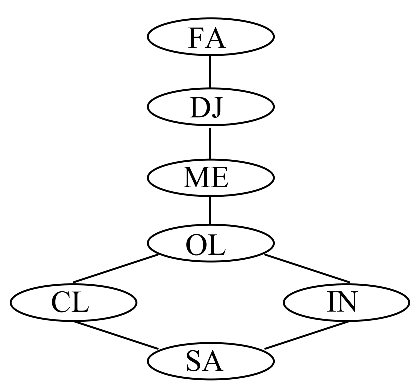


Figure 6. Similarity graph of positional operators

Table 2. Similarity distance between positional operators

	FA	DJ	ME	OL	CL	SA	IN
FA	0	1	2	3	4	5	4
DJ	1	0	1	2	3	4	3
ME	2	1	0	1	2	3	2
OL	3	2	1	0	1	2	1
CL	4	3	2	1	0	1	2
SA	5	4	3	2	1	0	1
IN	4	3	4	3	2	1	0

3.4 Indexing scheme with Ranking

Our schemes can support ranking for providing the retrieved results in the order of their similarity to a user query. For this, we devise a new weighting scheme which can rank the results based on the angle made between two objects and their positional operator. Let X be a spatial string in an iconic image and Y be a spatial string in an iconic query image. When two spatial strings X and Y are denoted by X=(A, B, R^{AB}, Ang) and Y=($QA, QB, QR^{AB}, QAng$) respectively, the following formula (1) calculates a weighting value between X and Y ($0 < W^{XY} \leq 1$) if both conditions are satisfied. Otherwise, W^{XY}= 0.

(i) the two objects from each string are the same, i.e., $A = QA$ and $B = QB$, and

(ii) the angle difference, i.e., |Ang-Qang|, is less than a threshold (θ).

$$W^{XY} = \left(1 - \frac{|Ang - QAng|}{\theta}\right) * \delta \qquad (1)$$

$$\text{where},$$

$$\delta = \frac{1}{1 + \text{sim\_dis}(R^{AB}, QR^{AB})} \quad \text{if sim\_dis}(R^{AB}, QR^{AB}) \leq \text{thres\_dis}$$

$$\delta = 0 \quad \text{otherwise}$$

Here $sim\_dis(R^{AB}, QR^{AB})$ means the similarity distance between R^{AB} and QR^{AB} and *thres_dis* means a threshold distance for deciding whether or not we consider the similarity distance between R^{AB} and QR^{AB}. That is, if $sim\_dis(R^{AB}, QR^{AB})$ is greater than *thres_dis*, its position weight, δ, equals to 0. Because an iconic image commonly has multiple objects, the number of spatial strings for an image is calculated as $m(m-1)/2$, where m is the number of objects in an iconic image. Therefore, we can compute a weight W_{PQ} between an iconic image (P) in the database and an iconic query image(Q) as the following formula (2). Here, n is the number of spatial strings in Q and W_i^{XY} is the weighting value between the i-th spatial string of Q and its matching spatial string of P.

$$W_{PQ} = \frac{\sum_{i=1}^{n} W_i^{XY}}{n} \qquad (2)$$

4. Inversion-based Access Method

Since the 9DLT and the SMR schemes make use of a signature file [11] as their access methods, they have good performance in terms of storage utilization. However, due to the characteristics of the signature file, they have two disadvantages. First, they can not support ranking for the retrieved result. Secondly, since they do post scanning, it takes long time in a retrieval operation, especially when the number of spatial strings are large. Thus, our schemes use an inverted file [12] as its access method in order to support ranking and fast retrieval. Figure 7 shows the structure of an inversion-based retrieval method used for our schemes.

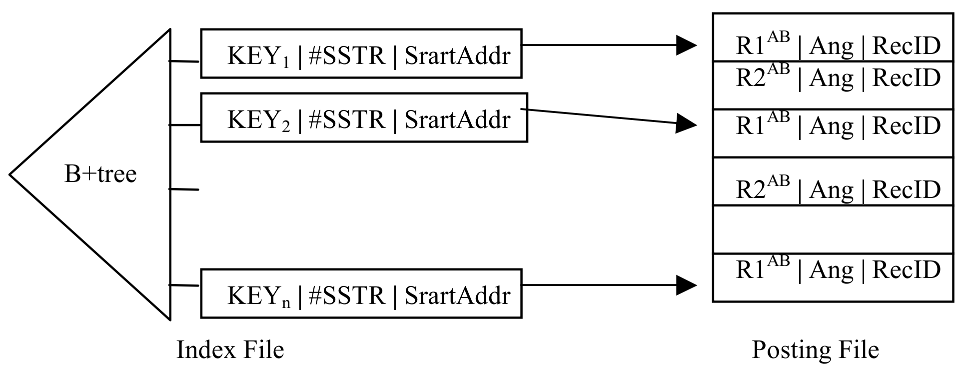


Figure 7. The inverted file structure used for our schemes

To construct a key, we combine two object identifiers from a spatial string. For instance, suppose a spatial string (*A, B, OL, 234*°). We concatenate A and B to form a string 'AB', which is used as the key of the string. In Figure 7, *recAddr* means the address of posting file record corresponding to a given key in the posting file and *#SSTR* means the number of the spatial strings with the key. In addition, R^{AB} indicates the topological operator of a spatial string, *Ang* indicates the angle of a spatial string, and *ImageID* indicates the identifier of an iconic image related with a spatial string.

4.1 Insertion

For inserting a spatial string being (A, B, R^{AB}, Ang) in a iconic image referred by *ImageID*, our scheme makes a key out of two object identifiers in the spatial string and search the key in our inversion-based retrieval method. If the key, i.e., 'AB', already exists in the index file, we read the posting file record pointed by the key and increase *#SSTR* by 1. Then, we insert R^{AB}, *Ang* , and *ImageID* into the posting file record. If the key does not exist in the index file, we make a new posting file record and set *#SSTR* to 1. Then, we insert R^{AB}, *Ang* , and *ImageID* into the posting file record. Finally, we insert the new key into the index file. Algorithm 1 shows the insertion algorithm of our scheme.

[*Algorithm 1*] Insertion Algorithm

Input : a set of spatial strings being (*A, B,* R^{AB}, *Ang*) in an iconic image referred *ImageID*

Output : *SUCCESS* or *FAIL*

Variables :

n : the number of spatial strings in an iconic image

key_i : key made from the *i-th* spatial string

recAddr : posting file record address with key_i in the index file

postingAddr : posting file record address after append or insertion is performed to(into) a posting file record

Begin:

```
for (i = 0; i < n; i++)
        if((recAddr = index_file_find(key_i)) != NULL)        {
                read__posting_record (recAddr);
                postingAddr = append(R^AB, Ang, lmageID);
                if(postingAddr != recAddr)
                        index_file_update(key_i , postingAddr);
        }
        else        {
                get_posting_record ();
                postingAddr = append(R^AB, Ang, lmageID);
                index_file_insert(key_i , postingAddr);
        }
```

End :

4.2 Retrieval

For retrieval, we first make a set of spatial strings, being (*QA, QB,* QR^{AB}, *QAng*), from an iconic query image. Then, we construct a key from each spatial string and search the index file with the keys. If all of the keys do not exist in the index file, there is no iconic image satisfying the query in the database. Meanwhile, if one of the keys, i.e., '*QAQB'* is found in the index file, we read the posting file record with the key from the posting file. Then, we obtain a pair of W^{XY} *and ImageID* by comparing (QR^{AB}, *QAng*) with all of the spatial strings in the posting file record. Next, we compute W_{PQ} by averaging W^{XY} for each *ImageID* after finding a set of W^{XY} *and ImageID* pairs for all the spatial strings of the query image. Thirdly, we obtain the retrieved results by excluding iconic images having smaller W_{PQ} than a given value. We finally rank the results in the order of W_{PQ} and provide them to users. Algorithm 2 shows the retrieval algorithm of our scheme.

[*Algorithm 2*] Retrieval Algorithm

Input : a set of spatial strings, being (*QA, QB,* QR^{AB}, *QAng*), in an iconic query image

Output : a list of { *ImageID,* W_{PQ}}

```
Variables :
        n : the number of spatial strings in an iconic query image
        k : the number of spatial strings in a posting file record
        key_i : key made from the i-th spatial string
        recAddr : posting file record address with key_i in the index file
        W^XY : weighting value between a spatial string in an iconic image and one in an iconic query image
        W_PQ : weighting value between an iconic image in the database and an iconic query image
        LIST : output list of {ImageID, W_PQ}
        N_FAIL : the number of cases in which key_i do not exist in the index file
Begin:
        N_FAIL = 0;
        for (i = 0; i < n; i++)
                if(recAddr = index_file_find(key_i)) == NULL) N_FAIL++;
                else    {
                        read_posting_record(recAddr);
                        for( j = 0; j < k; j++)
                            compute W^XY and get ImageID;
                        obtain a set of {ImageID , W^XY} pairs;
                }
        if(N_FAIL < n)   {
                calculate W_PQ by averaging W^XY for each ImageID;
                for (each ImageID )
                        if(W_PQ > thres )
                                store {ImageID ,W_PQ} into LIST;
                sort(LIST);
        }
End:
```

4.3 Example

When we have an iconic query image as shown in Figure 8, we determine the spatial relationships between two object A and B as follows:

- SRC scheme

  Spatial relation between two objects (R^{AB}) : OL
- SRR scheme

  Spatial relation on X-axis (R_X^{AB}) : *ME*

  Spatial relation on Y-axis (R_Y^{AB}) : *OL*

Spatial relation between two objects (R^{AB}) : *ME*

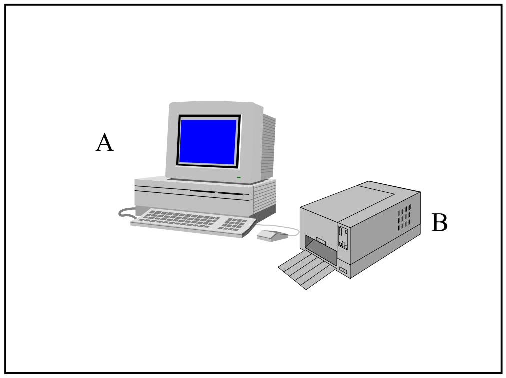


Figure 8. Iconic query image

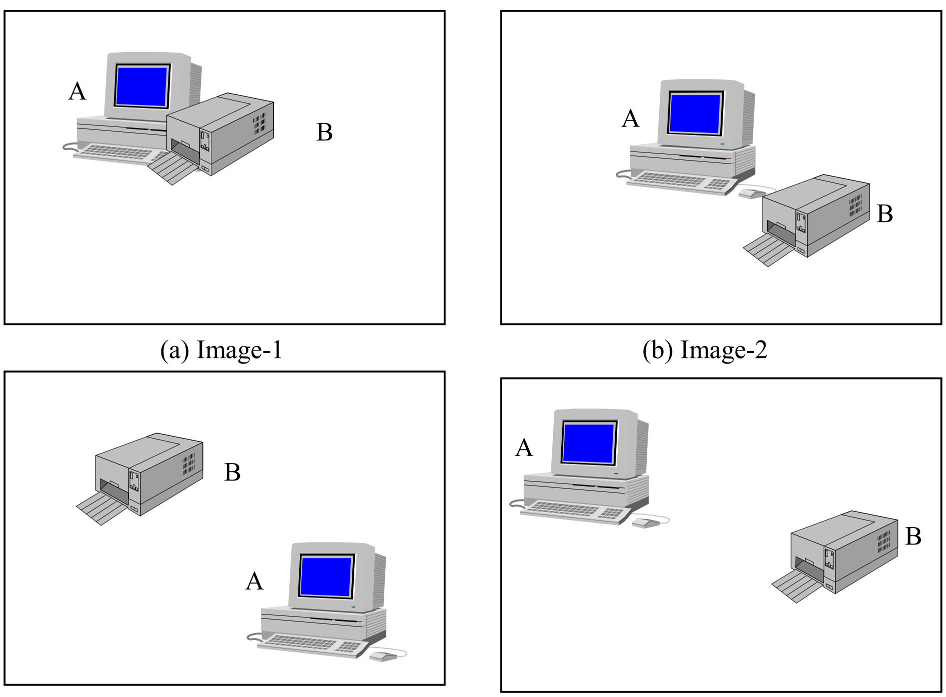
Therefore, the spatial string of the iconic query image in our schemes can be described as follows:

- SRC scheme

  (Q-Image) (*A, B, OL, 287*^{o})

- SRR scheme

  (Q-Image) (*A, B, ME, 287*^{o})



(c) Image-3

(d) Image-4

Figure 9. Example iconic images

Suppose we have four iconic images consisting of two icon objects A and B as shown in Figure 9. We generate the spatial string of each iconic image as follows:

- SRC scheme
  (Image 1) (*A, B, OL,* 309^{o})
  (Image 2) (*A, B, OL,* 278^{o})
  (Image 3) (*A, B, DJ,* 148^{o})
  (Image 4) (*A, B, FA,* 278^{o})
- SRR scheme
  (Image 1) (*A, B, OL,* 309^{o})
  (Image 2) (*A, B, ME,* 278^{o})
  (Image 3) (*A, B, DJ,* 148^{o})
  (Image 4) (*A, B, FA,* 278^{o})

By using the formula (1), we calculate W_{PQ} between the iconic query image and the example four images in the database. Here, *thres_dis* is assumed to be one. In the SRC scheme, the weighting values W_{PQ} of the images 1, 2, 3, and 4 to the iconic query image are 50.4%, 80%, 0%, and 40%, respectively. If we acquire as the retrieved result the iconic images having W_{PQ} greater than 50%, the retrieved result contains the images 1 and 2, so the images 3 and 4 being excluded from the results. If we want iconic images which have W_{PQ} greater than 30%, the images 1, 2 and 4 become the retrieved result. Here, since W_{PQ} of the image 2 is the largest and W_{PQ} of the image 4 is the smallest, the query results are provided to a user in the order of the image 2, the image 1, and the image 4. In the SRR scheme, the weighting values W_{PQ} of the images 1, 2, 3, and 4 to the iconic query image are 25.2%, 80%, 0%, and 40%, respectively. If we acquire as the retrieved result the iconic images having W_{PQ} greater than 50%, the retrieved result contains only the image 2, so the images 1, 3, and 4 being excluded from the results. If we want iconic images having W_{PQ} greater than 30%, the images 2 and 4 become the retrieved result. In the case, since W_{PQ} of the image 2 is larger than that of image 4, the query results are provided to a user in the order of the image 2 and the image 4.

5. Performance Analysis

For our experiment, we assume that an iconic image consists of icon objects each of which has its object identifier and its position. The position is described by two points in the XY-coordinates, such as the low left corner and the upper right corner. In addition, the threshold of angle difference , i.e., θ, is determined to be 45^{o} because many people use eight directions, such as north, northeast, as so on. *thres_dis* is also determined to be one because adjacent topological operators in their similarity graph are definitely considered to be similar. Finally, we make use of the following data.

- For the database used for our experiment, we collect 150 images from interior design books.
- The iconic image transformed from each image has 3 to 20 icon objects.
- The icon objects used in the database are 53 different types.

- For the user query, we generate 45 iconic query images.
- Each iconic query image has 2 to 3 icon objects.

For the performance evaluation, we compare our SRC and SRR schemes with the 9DLT and the SMR in terms of retrieval effectiveness as well as system efficiency. In order to evaluate the retrieval effectiveness, we make use of recall and precision measures [13]. Let *IRT* be the number of iconic images retrieved by a given query, *IRL* be the number of iconic images relevant to the query, and *IRR* be the number of relevant iconic images retrieved. To compute IRL, we make a test panel which finds relevant images manually from the database. The test panel is composed of ten graduate school students from our computer engineering department. Table 3 shows the *IRT, IRL* and *IRR* values of the 9DLT, the SMR, and our schemes. Here, the 9DLT(E) and the 9DLT(A) stand for the exact match and the approximate match of the 9DLT scheme, respectively. Similarly, the SMR(E) and the SMR(A) stand for the exact match and the approximate match of the SMR scheme. The SRC(40) and SRR(40) stand for our SRC and SRR schemes having W_{PQ} greater than 0.4, respectively. Equation (3) shows the formulae for precision and recall measures.

$$Recall = \frac{IRR}{IRL} * 100, \quad Precision = \frac{IRR}{IRT} * 100 \qquad (3)$$

Table 3. Average *IRT, IRL*, and *IRR* values

	IRT	*IRR*	*IRL*
9DLT (E)	21.38	8.33	10.09
9DLT (A)	21.38	8.33	10.09
SMR (E)	7.16	4.40	10.09
SMR (A)	8.44	5.18	10.09
SRC (40)	24.78	8.84	10.09
SRC (50)	15.60	6.00	10.09
SRC (60)	12.91	5.64	10.09
SRC (70)	11.13	5.27	10.09
SRR (40)	20.89	8.76	10.09
SRR (50)	14.07	6.98	10.09
SRR (60)	11.98	6.71	10.09
SRR (70)	9.71	6.24	10.09

For the precision and the recall, we adopt the 11-point measure [13]. Since the 9DLT and the SMR can not support ranking, we measure the performances in two ways for fair comparison, as shown in Table 4. One is for the ranking where the retrieved result is provided in the order of its relevance to a query. The other is for the non-ranking where the retrieved result is provided without ordering. In the case of the ranking, since the 9DLT and the SMR do not support the ranking, we rank the retrieved images in the order of the sequence retrieved. Table 4 shows the precision and recall of the 9DLT, the

SMR, our SRC, and our SRR. In the case of the non-ranking, our SRR scheme is nearly the same as the 9DLT, but it is superior to the SMR in terms of the recall while it is inferior to the SMR in terms of the precision. In addition, our SRC scheme is lower than the 9DLT and the SMR in terms of the recall. In the case of the ranking, our SRR and SRC schemes are superior to the 9DLT and the SMR in terms of both the precision and the recall. Our SRC scheme has a good result in terms of the precision, while our SRR scheme has a good result in terms of the recall. That is, our SRC scheme holds about 7% higher recall and 16-19% higher precision, while our SRR scheme holds about 25% higher recall and 9-12% higher precision. Figure 10 shows the recall-precision graph of the 9DLT(A), the SMR(A), the SRC(40) and the SRR(40) with the 11-point measure. As shown in the graph, our SRC and SRR schemes show overall better performance than the 9DLT and the SMR.

Table 4. Comparison of retrieval effectiveness

Retrieval Effectiveness	Non-Ranking		Ranking	
	Recall	Precision	Recall	Precision
9DLT (E)	71.3675	31.3010	26.9559	34.5807
9DLT (A)	71.3675	31.3010	26.9559	34.5807
SMR (E)	46.9810	56.3167	25.0000	30.8750
SMR (A)	62.4083	50.2611	26.6131	31.7019
SRC (40)	46.2806	30.1548	31.9964	59.0410
SRC (50)	36.9437	38.5424	39.7960	50.0905
SRC (60)	36.6734	41.1227	32.4071	40.3550
SRC (70)	43.6774	49.8143	28.1036	42.4512
SRR (40)	69.0556	36.3170	51.7525	43.4000
SRR (50)	57.6953	46.3286	58.2265	38.0101
SRR (60)	54.1238	47.3750	59.5278	35.4008
SRR (70)	49.4632	50.7071	58.7405	35.3712

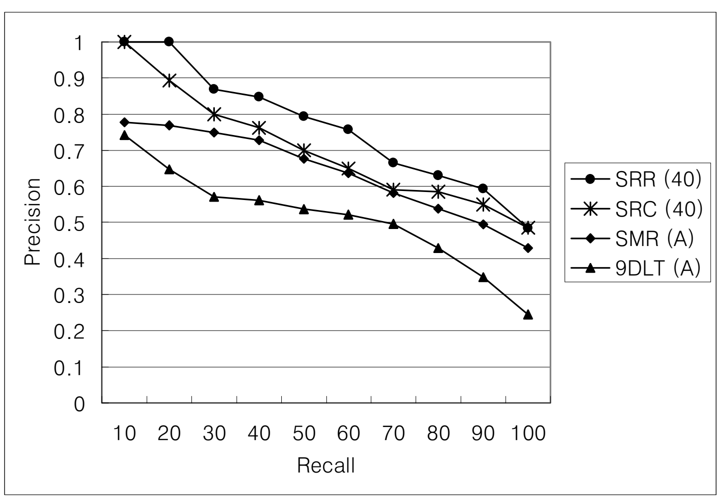


Figure 10. Recall-precision graph

In order to evaluate system efficiency, we compare our SRC and SRR schemes with the 9DLT and the SMR in terms of storage overhead (SO), insertion time, and retrieval time. SO is computed as *(index size / iconic image size)*100*. Table 5 shows the insertion, the retrieval time and SO. As shown in Table 5, the insertion operation of our SRC is about three times slower than that of 9DLT(E) and about twice slower than SMR(E). And our SRR is about five times slower than that of 9DLT(E) and about three times slower than SMR(E). This is because our SRC and SRR schemes require append operations to the posting file more frequently than the 9DLT(E) and the SMR(E). Meanwhile, the insertion operation of our SRC is three times faster than that of 9DLT(A) and about seven times faster than that of SMR(A). Similarly, our SRR is twice faster than the 9DLT(A) and about five times faster than SMR(A). This is because the 9DLT(A) and the SMR(A) take long time for making their signature files due to a large number of spatial strings.

Table 5. Performance comparison in terms of system efficiency

Efficiency	9DLT(E)	9DLT(A)	SMR(E)	SMR(A)	SRC	SRR
Insert Time(sec)	6.46	67.11	13.18	168.97	24.40	34.50
Retrieval Time(sec)	5.62	41.94	8.52	90.99	0.69	1.37
SO	19	63	24	90	471	651

The retrieval operation of our SRC is about eight times faster than the 9DLT(E), about sixty times faster than SMR(E), about twelve times faster than 9DLT(A), and about one hundred thirty times faster than SMR(A). And our SRR is about five times faster than the 9DLT(E), about seven times faster than SMR(E), about thirty times faster than 9DLT(A), and about seventy times faster than SMR(A). This is because our SRC and SRR schemes need not do post scanning because of using the inverted file. However, since our SRC and SRR schemes use the inverted file, they require about five to thirty times lager storage overhead than the 9DLT and the SMR. In addition, because the representation of our SRC scheme is simple than that of our SRR scheme, our SRC scheme is faster and takes little disk space than our SRR scheme.

6. Conclusions

In this paper, we proposed new spatial-match iconic image representations, called SRC and SRR, which combine the directional operators with the positional operators so that they might provide precise relationships between icon objects in an iconic image. Since our SRC and SRR schemes also support ranking for the retrieved result, they can rank the result in the order of their relevance to a user query. From our experiment, we showed that our SRC and SRR schemes should be superior to the 9DLT and the SMR in terms of retrieval effectiveness. In addition, our SRC and SRR schemes showed about six to one hundred thirty times faster retrieval efficiency than the 9DLT and the SMR, while they spent about eight to ten times more storage overhead. When we compare our SRC with our SRR, our SRR scheme is better in terms of retrieval effectiveness, while our SRC scheme is better in terms of system efficiency. Therefore, our SRC and SRR schemes are shown to be suitable for multimedia information systems which require ranking as well as fast retrieval. For future work, it is necessary to prove the superiority of our spatial-match iconic image representations by applying them to real multimedia database applications with a large amount of data.

References

[1] V.N. Gudivada and V.V. Raghavan, "Design, and Evaluation of Algorithms for Image Retrieval by Spatial Similarity," ACM Transactions on Information Systems, Vol. 13, No. 2, pp.115-144, 1995.

[2] S. K. Chang, Q.Y. Shi, and C.W. Yan, "Iconic Indexing by 2D strings", IEEE Transaction on Pattern Recognition and Machine Intelligence, 9(3): pp. 413-428, 1987.

[3] S.Y. Lee and F.J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images using 2D C-string Knowledge Representation," Pattern Recognition, Vol. 25, No. 3, pp. 305-318, 1992.

[4] C.C. Chang and J.H. Jiang, "A Fast Spatial Match Retrieval Using a Superimposed Coding Techniques," International Symposium on ADTI(Nara, Japan), pp.71-78, 1994.

[5] J.W. Chang, Y.J. Kim and K.J. Chang," A Spatial Match Representation Scheme for Indexing and Querying in Iconic Image Databases," ACM International Conference on Information and Knowledge Management, pp. 169-176, Nov. 1997.

[6] M. Nabil, A.H. Ngu, and J. Shepherd, "Picture Similarity Retrieval Using the 2D Projection Interval Representation," IEEE Transaction on Knowledge and Data Engineering, Vol. 8, No. 4, 1996.

[7] T. Tagashira and T. Amagasa and M. Aritsugi and Y. Kanamori, "Interval-Based Representation of Spatio-Temporal Concepts," 9th International Conference on Advanced Information Systems Engineering (CaiSE*97) Springer LNCS 1250, pp. 231-244, 1997.

[8] J. Halpern and Y. Shoham, "A Prepositional Model Logic of Time Intervals", Journal of Association Computing Machinery, Vol. 33, pp.935-962, Oct. 1991.

[9] M. Safar and C. Shahabi, "2D Topological and Directional Relations in the World of Minimum Bounding Circles," International Database Engineering and Applications Symposium, pp. 239-247, 1999.

[10] D. Papadias, Y. Theodoridis, T. Sellis and M.J. Egnhofer, "Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees," ACM SIGMOD on Management of Data, pp. 92-103, 1995.

[11] C. Faloutsos and S. Christodoulakis, "Signature Files : An Access Methods for Documents and its Analytical Performance Evaluation, " ACM Transaction on Database Systems, 2(4) pp. 267-288, 1984.

[12] G. Salton, and M. McGill, "An Introduction to Modern Information Retrieval," McGraw-Hill, 1983.

[13] R.R. Korfhage, "Information Storage and Retrieval," Wiley Computer Publishing, pp.199-202, 1997.