A Spatial Match Representation Scheme for Indexing and Querying in Iconic Image Databases

Jae-Woo Chang   Yeon-Jung Kim   Ki-Jin Chang

Department of Computer Engineering, Chonbuk National University
Chonju, Chonbuk 560-756, South Korea
{jwchang, yjkim}@dblab.chonbuk.ac.kr

Abstract

In multimedia information retrieval applications, content-based image retrieval is essential for retrieving relevant multimedia documents. The purpose of our paper is to provide both effective representation and efficient retrieval of images when a pixel-level original image is automatically or manually transformed into its iconic image containing meaningful graphic descriptions, called icon objects. For this, we propose a new spatial match representation scheme to describe spatial relationships between icon objects precisely by using accurate positional operators as well as by expressing objects as rectangles, rather than pointer. In order to accelerate image searching, we also design an efficient retrieval method using a signature file technique. Finally, we show from our experiments that our representation scheme achieves better retrieval effectiveness than the 9-DLT scheme.

1 Introduction

Recently, much attention has been paid to Multimedia Information Retrieval(MIR) because we have had so many applications that should be supported by handling multimedia data, such as text, image, video, audio, and animation. The applications include digital libraries, advertisements, medical information, remote sensing and astronomy, cartography, digital newspapers, and architectural design. So far, text attributes in multimedia documents have mainly been used for supporting queries by content. The approach using text content(e.g., captions and keywords) has a couple of problems. First, the original keywords do not allow for unanticipated searching. The other problem is that the caption is not adequate to describe the layout, sketch, and shape of the image. Therefore, in order to support MIR applications effectively, content-based image retrieval is essential because it plays an important role in retrieving relevant multimedia documents.

Given a pixel-level original image, various image processing and understanding techniques are used to identify domain 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 database. Moreover, this task may be carried out in a semi-automated way or in an automated way, depending on the domain and complexity of the images. An iconic image is obtained by associating each domain object of the original image with a meaningful graphic description, called an icon object. Thus, an iconic image representation can provide users with a high level of image abstraction. The iconic image representation has some advantages. First, the use of iconic images avoids the need for 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 MIR applications. Secondly, the iconic image representation is useful in a distributed database environment where an original image is stored only at a central node and its iconic image is stored at each local node.

In the paper, we assume that all images at the pixel level are analyzed prior to storage so that icon objects can be extracted from their content and stored into the database together with the original images. The icon objects are used to search the image database and to determine whether an image satisfies query selection criteria. Ultimately, the effectiveness of MIR systems depends on the type and correctness of image content representation, the type of queries allowed, and the efficiency of search techniques designed. The purpose of our paper is to provide both effective representation and efficient retrieval of images when a pixel-level original image is automatically or manually transformed into its iconic image including icon objects. For this, we propose a new spatial match representation scheme to support the content-based image retrieval in an effective way. The proposed representation scheme can describe spatial relationships between icon objects in a precise way because it represents the icon objects as rectangles rather than as points,

CIKM 97 Las Vegas Nevada USA

and it makes use of accurate positional operators. In order to accelerate image searching, we also design an efficient retrieval method using a signature file technique.

The remainder of this paper is organized as follows. A review of related work done in the area of iconic image databases is introduced in Section 2. The proposed spatial match representation scheme is described in Section 3. A new efficient retrieval method to accelerate image searching is presented in Section 4. Section 5 provides a comparison of the proposed representation scheme with the conventional 9-DLT. Finally, Section 6 concludes the paper with some issues for future research.

2 Related Work

There have been many proposals for spatial match representation and retrieval in order to search symbolic images efficiently, satisfying certain spatial relationships [1, 2, 3, 4]. In particular, there have been two previous efforts on spatial match retrieval using signature file techniques, namely the 2D(Dimensional)-string based scheme [2] and the 9DLT(Direction Lower Triangular) based scheme [4].

2.1 The 2D-string based scheme

Chang, Shi and Yan [1] first proposed a 2D string to represent symbolic images. The 2D string makes use of a symbolic projection to represent a symbolic image by preserving some spatial knowledge of objects embedded in an original image. In addition, they defined three types (type-0, type-1, and type-2) of 2D sequence pattern matching. Lee and Shan [2] proposed a 2D-string based scheme to express some types of spatial relationships of symbolic images. In this scheme, they generated four kinds of two-level signature files by associating each symbolic image with a record signature and by relating some images with a block signature. For convenience of signature generation, they defined a spatial string to represent the pairwise spatial relationships embedded in a 2D string. A type-i 1D spatial character V^{AB} is a character describing the spatial relationship between A and B symbols in the 1D string as follows:

$$
\begin{array}{lll}
\text{(type-0)} & V^{AB} = "0" & \text{if } r(A) = r(B) \\
 & V^{AB} = "0" \text{ and } "1" & \text{if } r(A) < r(B) \\
 & V^{AB} = "0" \text{ and } "2" & \text{if } r(A) > r(B) \\
\text{(type-1)} & V^{AB} = "0" & \text{if } r(A) = r(B) \\
 & V^{AB} = "1" & \text{if } r(A) < r(B) \\
 & V^{AB} = "2" & \text{if } r(A) > r(B) \\
\text{(type-2)} & V^{AB} = "0" + str(r(A) - r(B)) & \text{if } r(A) = r(B) \\
 & V^{AB} = "1" + str(r(B) - r(A)) & \text{if } r(A) < r(B) \\
 & V^{AB} = "2" + str(r(A) - r(B)) & \text{if } r(A) > r(B)
\end{array}
$$

Here $r(X)$ is the rank of symbol X, "+" denotes the string concatenating operator, and str(X) is a transformation function from integer to string; for example, str(3)="3". A type-i 2D spatial string for symbols A and B when i=0, 1, and 2, S_i^{AB}, is a string formed by concatenating A, B, and type-i spatial characters V_X^{AB} and V_Y^{AB}, where V_X^{AB} is a spatial character along the X-axis and V_Y^{AB} is a spatial character along the Y-axis. Therefore, S_i^{AB} is written as $A + B + V_X^{AB} + V_Y^{AB}$. S_i is a set of S_i^{AB} for all pairs of symbols A and B in an symbolic image.

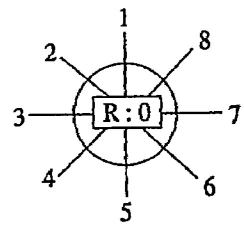

Figure 1: 9DLT direction codes

2.2 The 9DLT based scheme

Chang [3] proposed a 9DLT representation to describe the type-1 spatial relationship embedded in a 2D string. In this representation, nine integers(i.e., 1, 2, 3, 4, 5, 6, 7, 8, and 0) are used to represent pairwise spatial relationships embedded in a 2D string. Figure 1 shows the nine direction codes. Chang and Jiang [4] proposed a 9DLT based scheme to express three types of spatial strings by extending the 9DLT representation so that they can fully support the description of type-0, type-1, and type-2 pairwise spatial relationships embedded in a 2D string. They also designed a quick-filter based signature file organization as a filter for spatial match retrieval of images. The 9DLT based scheme describes a spatial representation between A and B symbols as follows.

$$
\begin{array}{lll}
\text{(type-0)} & ST_0^{AB} = (A, B, D'_{AB}) & \\
 & D'_{AB} = 0 & \text{if } D_{AB} = 0 \\
 & D'_{AB} = 0 \text{ and } 1 & \text{if } D_{AB} = 1 \\
 & D'_{AB} = 0 \text{ and } 3 & \text{if } D_{AB} = 3 \\
 & D'_{AB} = 0 \text{ and } 5 & \text{if } D_{AB} = 5 \\
 & D'_{AB} = 0 \text{ and } 7 & \text{if } D_{AB} = 7 \\
 & D'_{AB} = 0{,}1{,}2 \text{ and } 3 & \text{if } D_{AB} = 2 \\
 & D'_{AB} = 0{,}3{,}4 \text{ and } 5 & \text{if } D_{AB} = 4 \\
 & D'_{AB} = 0{,}5{,}6 \text{ and } 7 & \text{if } D_{AB} = 6 \\
 & D'_{AB} = 0{,}1{,}7 \text{ and } 8 & \text{if } D_{AB} = 8 \\
\text{(type-1)} & ST_1^{AB} = (A, B, D_{AB}) & \\
\text{(type-2)} & ST_2^{AB} = (A, B, D_{AB}, SC_X^{AB}, SC_Y^{AB}) & \\
 & SC_X^{AB} = 0 \text{ if } |rx(A) - rx(B)| \le 1 & \\
 & SC_X^{AB} = 1 \text{ if } |rx(A) - rx(B)| > 1 & \\
 & SC_Y^{AB} = 0 \text{ if } |rx(A) - rx(B)| \le 1 & \\
 & SC_Y^{AB} = 1 \text{ if } |rx(A) - rx(B)| > 1 &
\end{array}
$$

Here, S_i^{AB} represents the type-i spatial strings for A and B symbols, and (A, B, D_{AB}) denotes the 9DLT representation of symbols A and B. SC_X^{AB} and SC_Y^{AB} represent the spatial codes for symbols A and B in the X-axis and the Y-axis, respectively. Expression $|t|$ denotes the absolute value of t; for example, $|-2| = 2$.

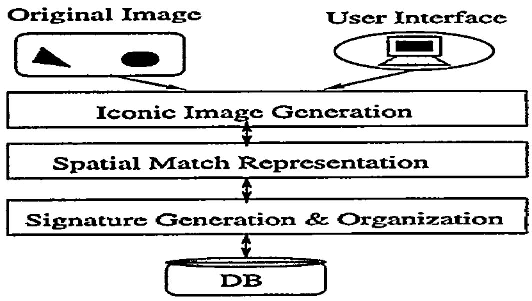

Figure 2: The architecture of a spatial match retrieval system

3 A New Spatial Match Representation Scheme

For image indexing, a large number of known image processing and understanding techniques [5] can first be used to identify some domain objects and their relationships in an original image. Next, we can easily obtain an iconic image by associating a meaningful icon object with each domain object in the original image. By using some spatial match representations, 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 image in the same way as that used in the image indexing. Next, we can represent the query iconic image as spatial strings by using some spatial match representations. Then, we can generate a query signature from the spatial strings and can get some potential matches by comparing the query signature with all of the signatures in the signature file. By excluding some false matches from the potential ones, we can finally retrieve some iconic images to satisfy the user query. The architecture of a spatial match retrieval system is shown in Figure 2.

For spatial match representations, there have been two main representation schemes to search image results efficiently, satisfying certain spatial relationships [1, 3]. However, both representation schemes have a critical problem in that they represent each domain object of an original image as its icon object being expressed as a point. As a result, they are not accurate enough to express spatial relationships between objects for handling original images with a complex scene. Therefore, we propose a new spatial match representation scheme to support effective content-based image retrieval. The proposed representation scheme can accurately describe spatial relationships between icon objects of iconic images because it represents each icon object as a rectangle, rather than as a point.

3.1 Spatial match representation scheme

A generic scene S of an original image is defined as a set of icon objects O in its iconic image. Therefore, an iconic description of the scene is a set of spatial relationships between pairs of icon objects. The spatial relationship(SR) is expressed as $SR = p\ O_p\ q$, where p and q are the projections over the X-axis (or Y-axis) of A and B objects, respectively, and O_p is a positional operator, which relates the intervals originated by the projections of p and q on the X-axis (or Y-axis). For new positional operators, we can extend some operators used for the specification of temporal relationships between time intervals in interval logic [6]. Thus, we propose new fifteen positional operators to express all of the possible relationships between a pair of intervals. In the case of the X-axis, the projections of p and q are referred to as $p = [x_{p1}, x_{p2}]$, and $q = [x_{q1}, x_{q2}]$, respectively, where $x_{p1} < x_{p2}$ and $x_{q1} < x_{q2}$. Here "d" indicates the threshold value of distance between two objects. The value of d can be determined as the average length of reference intervals, and the optimal value should be determined by a large number of experiments.

(a) p far-away-after (>>) q iff $x_{p1} \geq x_{q2} + d$
(b) p strictly-after (>+) q iff $x_{q2} < x_{p1} < x_{q2} + d$
(c) p after with right adjacency (>=) q iff $x_{q2} = x_{p1}$
(d) p after (>) q iff $x_{q1} < x_{p1} < x_{q2}$ and $x_{q2} < x_{p2}$
(e) p is-included-by with left adjacency (>−) q
  iff $x_{q1} < x_{p1}$ and $x_{p2} = x_{q2}$
(f) p includes with right adjacency (>|) q
  iff $x_{q1} = x_{p1}$ and $x_{q2} < x_{p2}$
(g) p includes (><) q iff$x_{p1} < x_{q1}$ and $x_{q2} < x_{p2}$
(h) p spatial-coincidence (=) q iff $x_{p1} = x_{q1}$ and $x_{p2} = x_{q2}$
(i) p is-included-by (<>) q iff $x_{q1} < x_{p1}$ and $x_{p2} < x_{q2}$
(j) p includes with left adjacency (<|) q
  iff $x_{q1} > x_{p1}$ and $x_{q2} = x_{p2}$
(k) p is-included-by with right adjacency (<−) q
  iff $x_{q1} = x_{p1}$ and $x_{p2} < x_{q2}$
(l) p before (<) q iff $x_{q1} < x_{p2} < x_{q2}$ and $x_{p1} < x_{q1}$
(m) p before with left adjacency (<=) q iff $x_{p2} = x_{q1}$
(n) p strictly-before (<+) q iff $x_{p2} < x_{q1} < x_{p2} + d$
(o) p far-away-before (<<) q iff $x_{q1} \geq x_{p2} + d$

The visual sketch of positional operators used for our representation scheme are given in Figure 3. For the convenience of signature generation, we also define a spatial string to represent the pairwise spatial relationships between objects in a two-dimensional image. For this, we express two types of spatial strings so that they can support both the exact and the approximate match. First, an exact-match i-axis spatial character, DE_i^{AB}, is a character describing a spatial relationship between objects A and B when the projections of A and B in terms of the i-axis are referred to as $p = [x_{p1}, x_{p2}]$, and $q = [x_{q1}, x_{q2}]$, respectively, where $x_{p1} < x_{p2}$ and $x_{q1} < x_{q2}$. The exact-match spatial character is written as the following:

4 A New Efficient Retrieval Method

In order to support fast searching of spatial strings for iconic images, it is necessary to construct an efficient retrieval method using a signature file organization because of its main advantages: fast retrieval time and low storage overhead [7]. When an iconic image consists of both icon objects and a set of spatial strings among them, we first create an object signature for each object in the iconic image and superimpose all of the signatures by using a superimposed coding technique. Then, we create an approximate-match signature by superimposing all of the signatures, each of which is made from each approximate-match spatial string for the iconic image. In the same way, we also construct an exact-match signature for the iconic image. Superimposing signatures leads to reducing the disk space to be accessed dramatically. We finally construct an image signature by concatenating the object, the approximate-match, and the exact-match signatures by using a disjoint coding technique.

Therefore, we can offer a way to answer a variety of user queries effectively since an image signature is composed of three parts of signatures. For example, if a user query needs some image results, including icon objects A and B, we can access only a portion of object signatures, thus dramatically reducing the query processing time. Similarly, if a user query requires all relevant images satisfying a certain relationship approximately, we can access only a portion of approximate-match signatures to answer the query.

4.1 Signature generation

With a set of exact-match spatial relationship strings (ESRs) corresponding to a given iconic image, we can generate a set of approximate-match spatial relationship strings (ASRs) as well as an object list (OL). Given the OL, a set of ASRs, and a set of ESRs, we can also generate three kinds of signatures, i.e., the object, the approximate-match, and the exact-match signatures for the iconic image. Then, an image signature for the iconic image is constructed by concatenating these three signatures into one signature. The algorithm to generate an image signature is illustrated below.

[Algorithm 1] Generation of image signature
Input: a set of ESRs for an iconic image, each being (A,B, P_X^{AB},P_Y^{AB})
Output: image signature, IS
Variables:
S_{obj}, S_{app}, S_{exa} : object, approximate-match, and exact-match signature for an iconic image, respectively
so_k : object signature for the k-th object of the OL
sa_i, se_i : approximate-match and exact-match signature for the i-th ESR, respectively
s_i^j : approximate-match signature for the j-th ASR of the i-th ESR

Begin:
    $S_{obj} = 0$; $S_{app} = 0$; $S_{exa} = 0$;
    Compute the OL from a set of ESRs;
    while(each k-th object of the OL for some k) {
        Create so_k from the k-th object of the OL;
        $S_{obj} = S_{obj} \vee so_i$;
    } /* while loop for k */
    while(each i-th ESR for some i) {
      Create se_i from the i-th ESR; $S_{exa} = S_{exa} \vee se_i$;
        Determine a set of ASRs from the i-th ESR;
        $sa_i = 0$;
        while(each j-th ASR for some j) {
        Create s_i^j from the j-th APR; $sa_i = sa_i \vee s_i^j$;
        } /* while loop for j */
    $S_{app} = S_{app} \vee sa_i$;
    } /* while loop for i */
    $RS = S_{obj} || S_{app} || S_{exa}$;
End:

4.2 Insertion and Retrieval

When a set of signatures for an iconic image is generated using Algorithm 1, we can store the object signature, the approximate-match signature and the exact-match signature into an object signature file, an approximate-match one and an exact-match signature file, respectively. Therefore, the insertion of an image signature can be easily handled because it only needs to append its three parts of signatures to those three signature files.

When a user query is given, it can be transformed into a query signature using Algorithm 1. Depending on whether the query belongs to an approximate-match or an exact-match type, we can decide in what sequence three signature files should be accessed so that we may achieve good retrieval performance. After accessing the corresponding signature files, we can obtain some qualifying signatures to satisfy the relationship strings in the query. Finally, we can find iconic image results by examining whether the iconic images corresponding to the qualifying signatures actually satisfy the query. If necessary, we can retrieve some pixel-level original images given by the iconic image results. Both the insertion and retrieval algorithms are omitted because of their simplicity.

4.3 Example

We assume that we have four iconic images consisting of icon objects A and B as shown in Figure 4. A set of approximate-match and exact-match spatial relationship strings in our basic representation can be obtained as follows:

- exact-match representation
  (Image-1) (A,B,6,1)
  (Image-2) (A,B,5,1)
  (Image-3) (A,B,6,0)

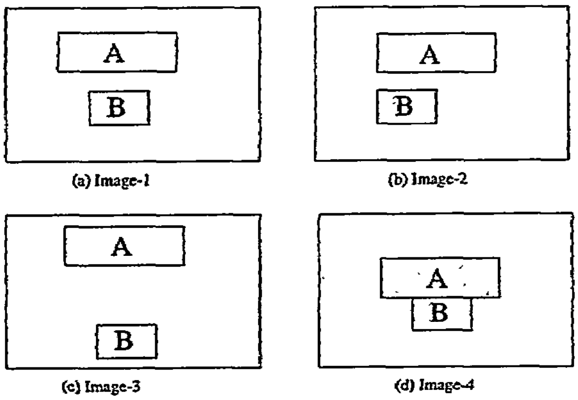

Figure 4: Four iconic images as example

(Image-4) (A,B,6,2)

- approximate-match representation
  (Image-1) (A,B,5,0),(A,B,5,1),(A,B,5,2),(A,B,6,0),(A,B,6,1),(A,B,6,2),(A,B,9,0),(A,B,9,1),(A,B,9,2)
  (Image-2) (A,B,3,0),(A,B,3,1),(A,B,3,2),(A,B,5,0),(A,B,5,1),(A,B,5,2),(A,B,6,0),(A,B,6,1),(A,B,6,2),(A,B,7,0),(A,B,7,1),(A,B,7,2)
  (Image-3) (A,B,5,0),(A,B,5,1),(A,B,6,0),(A,B,6,1),(A,B,9,0),(A,B,9,1)
  (Image-4) (A,B,5,1),(A,B,5,2),(A,B,5,3),(A,B,6,1),(A,B,6,2),(A,B,6,3),(A,B,9,1),(A,B,9,2),(A,B,9,3)

To create the signature of the four iconic images, we assume that the object signature has 6 bits in length, the approximate-match signature has 18 bits, and the exact-match signature has 12 bits. In addition, we assume that three hashing functions are used to generate these signatures, such as h_{obj}, h_{app}, and h_{exa}. Based on them, we can generate the signatures of the four images. Figure 5 illustrates a signature file structure after we insert the four image signatures. For example, suppose that we have a query to find such an iconic image as Image-Q in Figure 6. To answer this query, we first generate a set of spatial representation strings(SRS) for Image-Q in our representation scheme as follows:

- exact-match representation
  (A,B,6,1)
- approximate-match representation
  (A,B,5,0),(A,B,5,1),(A,B,5,2),(A,B,6,0),(A,B,6,1),(A,B,6,2),(A,B,9,0),(A,B,9,1) (A,B,9,2)

Next, we create the object, the approximate-match, and the exact-match signature of Image-Q as follows:

$$IS^{Q}_{obj} = 011011$$
$$IS^{Q}_{app} = 011100\ 001110\ 111000$$
$$IS^{Q}_{exa} = 000100\ 000100$$

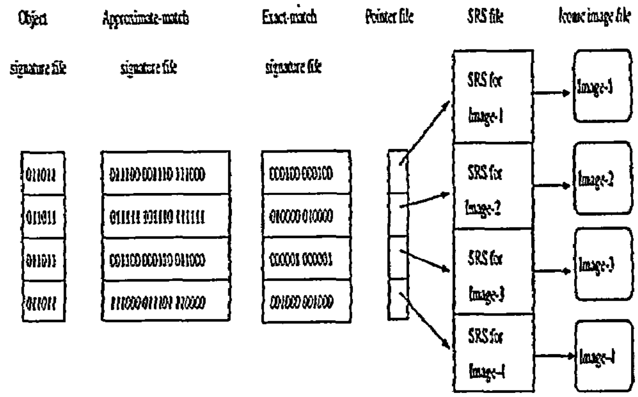

Figure 5: A signature file structure after inserting four iconic images

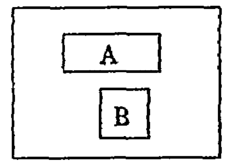

Figure 6: Image-Q: A query iconic image

Finally, if we require some exact-match answers, we can compare IS^{Q}_{exa} with the four signatures in the exact-match signature file. This leads to one qualifying signature because the first signature in the exact-match signature file satisfies the bit pattern of IS^{Q}_{exa}. Therefore, we obtain one qualifying iconic image, i.e., Image-1, because the ESR of image-1 actually contains the query's ESR. Also, if we require some approximate-match answers, we can search for only the four signatures of the approximate-match signature file. We obtain four qualifying signatures because they contain '1' bit in at least one of the bit positions, where IS^{Q}_{app} has bits set by one. Thus, we can access the ARSs of iconic images corresponding to the qualifying signatures so that we can find out some false drops. As an approximate-match answer, we finally obtain all of four iconic images because the ASRs of every iconic image include at least one of the ASRs of Image-Q.

5 Performance Evaluation

5.1 Our Experiments

We first assume that a pixel-level original image should be automatically or manually transformed into its iconic image including only icon objects, prior to storage into the database. This is because the purpose of our paper is to provide effective representation and fast searching of images after their icon objects are extracted by image analysis preprocessing. We also assume that an iconic image consists of icon objects, each having its icon name and its position. For our

#of icon objects in a query	9-DLT scheme		Our scheme	
	Approx.	Exact	Approx.	Exact
2	0.12	0.19	0.20	0.92
3	0.39	0.52	0.56	0.94
4	0.45	0.50	0.50	0.93
5	0.47	0.53	0.53	0.95
Precision	0.35	0.43	0.45	0.94

Table 1: Precision measure of our representation scheme

experiment, we generate the following iconic databases [8].

- Icon objects have 25 different types.
- An iconic image consists of two to ten icon objects.
- The total number of iconic images used for our experiment is 10,000.
- A query iconic image contains two to five icon objects.
- The number of queries for the experiment is 1,000.

In order to evaluate retrieval effectiveness [9], we make use of recall and precision measures. 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. The relevant images can be determined by computing the similarity between two iconic images, based on their spatial relationship. The recall and precision measures are computed as the following:

$$Recall = \frac{IRR}{IRL} * 100$$

$$Precision = \frac{IRR}{IRT} * 100$$

When a variety of queries are executed one thousand times, Table 1 shows the retrieval effectiveness of our representation scheme, in terms of precision measure. Here, "Exact" means a query type for the exact match and "Approx." means one for the approximate match. Our representation scheme achieves considerably better retrieval precision, compared to the 9DLT scheme. That is, it is shown that our scheme holds about 0.1 higher precision value in the exact match and about 0.5 greater value in the approximate match, while the recall value of our scheme is nearly the same as that of the 9-DLT one. When the number of icon objects in a query is small, it is shown that the precision values of the exact-match query are higher than those of the approximate-match one. As the number of icon objects is increased, the precision values of the approximate-match query are closer to those of exact-match one. This is because the number of qualifying iconic images is dramatically decreased as the number of objects in a query is increased.

Retrieval effectiveness	9-DLT scheme		Our scheme	
	Approx.	Exact	Approx.	Exact
Precision	0.2	0.38	0.30	0.94
Recall	1.0	1.0	1.0	1.0

Table 2: Retrieval effectiveness of our representation scheme

In order to verify the correctness of our experiment, we also implemented our representation scheme using 100 real interior design images. The iconic images corresponding to the real images were obtained by manual transformation. Iconic objects forming the iconic images have 20 different types related with an interior design field, such as, bed, chair, desk, sofa, table, armchair, standard lamp, bookcase, dressing table, wardrobe, oven, refrigerator, and so forth. A query iconic image contains two to three icon objects. When a variety of queries are executed ten times, Table 2 shows the retrieval effectiveness of our representation scheme, in terms of precision and recall measures. Our representation scheme shows approximately the same retrieval effectiveness results, compared to those in Table 1. In case of the approximate-match query, the precision value of our implemented scheme with real images is a little lower than that of our experiment with system-generated images. This is because we don't have a large number of real images for obtaining sufficient qualifying images to answer a given query. It is shown from the results that our representation scheme holds about 0.3 precision value in the approximate match and about 0.9 in the exact match, while their recall values are kept 1.0.

5.2 Analysis on our experiments

When a query have two icon objects, we can obtain one spatial relationship from the query. According to the relationship, user queries with icon objects A and B can be classified into the following seven cases. Here the projection of A and B objects over the X-axis are expressed as px and qx, respectively. Similarly, the projection of A and B objects over the Y-axis are expressed as py and qy. The fifteen positional operators can be classified into *away*, *intersect*, and *contain* groups, i.e., AW = $\{>>, >+, <+, <<\}$, IN = $\{>=, >, <, <=\}$, and CO = $\{>-, >|, ><, =, <>, <|, <-\}$.

(Case 1)px O_x qx and pyO_y qy,where $O_x \in$ AW,$O_y \in$ AW.
(Case 2)px O_x qx and pyO_y qy,where $O_x \in$ IN,$O_y \in$ AW.
(Case 3)px O_x qx and pyO_y qy,where $O_x \in$ AW,$O_y \in$ IN.
(Case 4)px O_x qx and pyO_y qy,where $O_x \in$ IN,$O_y \in$ IN.
(Case 5)px O_x qx and pyO_y qy,where $O_x \in$ CO,$O_y \in$ AW.
(Case 6)px O_x qx and pyO_y qy,where $O_x \in$ AW,$O_y \in$ CO.
(Case 7)px O_x qx and pyO_y qy,where $O_x \in$ AW,$O_y \in$ CO.

Figure 7 illustrates thirteen iconic images containing *window* and *standard lamp* icon objects and the label of

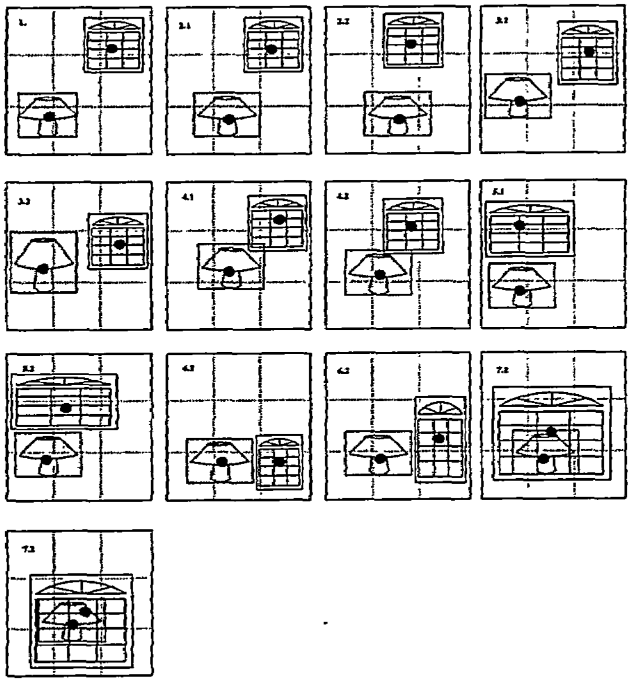

Figure 7: Iconic images in terms of seven cases

each iconic image means the case number. When we represent iconic images using the 9-DLT scheme and our representation scheme, we can obtain two query result sets for the following queries.

(Query 1) iconic image 1 - Case 1

- 9-DLT scheme: 1, 2.1, 3.1, 4.1, 5.2, 6.2
- Our scheme: 1

(Query 2) iconic image 2.1 - Case 2

- 9-DLT scheme: 1, 2.1, 3.1, 4.1, 5.2, 6.2
- Our scheme: 2.1, 2.2

(Query 3) iconic image 4.1 - Case 4

- 9-DLT scheme: 1, 2.1, 3.1, 4.1, 5.2, 6.2
- Our scheme: 4.1, 4.2

(Query 4) iconic image 5.1 - Case 5

- 9-DLT scheme: 2.2, 4.2, 5.1, 7.1
- Our scheme: 5.1, 5.2

(Query 5) iconic image 5.2 - Case 5

- 9-DLT scheme: 1, 2.1, 3.1, 4.1, 5.2, 6.2
- Our scheme: 5.1, 5.2

We show from the result sets of the given queries that our representation scheme achieves much better precision than the 9-DLT scheme. That is why our representation scheme describes spatial relationships between icon objects accurately by using precise positional operators as well as by expressing objects as rectangles, rather than as points. For example, if we make a query image to be the iconic image 2.1, we can obtain two qualifying images when using our scheme, while we get six qualifying images when using the 9-DLT scheme. Thus we can obtain more appropriate images to satisfy a query condition accurately when we represent iconic images by using our scheme.

6 Conclusions and Future work

We proposed our spatial match representation scheme so as to support content-based image retrieval in an effective way. Our representation scheme accurately described spatial relationships between icon objects because it could represent the icon object as a rectangle and make use of precise positional operators. To accelerate searching, we also designed a signature file organization where three signature files are constructed for storing object, exact-match, approximate-match signatures, separately. In order to prove the superiority of our representation scheme on retrieval effectiveness, we compared our schemes with the 9-DLT scheme, in terms of both precision and recall measures. We showed from our experiment that our representation scheme improved retrieval precision by about 0.5 in the exact match, and by up to 0.3 in the approximate match, compared with the 9-DLT scheme. For further work, our representation scheme should be applied to real application areas using iconic images, proving the efficiency of our scheme in these areas.

References

[1] S. K. Chang, Q. Y. Shi, and C. W. Yan. Iconic indexing by 2d strings. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 9(3):413–428, 1987.

[2] S. Y. Lee and M. K. Shan. Access methods of image databases. *International Journal of Pattern Recognition and Artificial Intelligence*, 4(1):27–44, 1990.

[3] C. C. Chang. Spatial match retrieval of symbolic pictures. *Information Science and Engineering*, pages 142–145, 1991.

[4] C. C. Chang and J. H. Jiang. A fast spatial match retrieval using a superimposed coding technique. In *Proc. of the Int'l Symposium on Advanced database Technologies and Their Integration*, pages 71–78, Nara, Japan, 1994.

[5] C. Faloutsos et al. Efficient and effective querying by image content. *Journal of Intelligent Information Sysyems*, 3:231–262, 1994.

[6] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of Association Computing Machinary*, 38:935–962, 1991.

[7] C. Faloutsos and S. Christodoulakis. Signature files: An access methods for documents and its analytical performance evaluation. *ACM Transactions on Database Systems*, 2(4):267–288, 1984.

[8] V. N. Gudivado. Tessa - an image testbed for evaluating 2d spatial similarity algorithms. *ACM SIGIR Forum*, pages 17–36, Fall 1994.

[9] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.