15 A THREE-DIMENSIONAL REPRESENTATION SCHEME FOR INDEXING AND QUERYING IN ICONIC IMAGE DATABASES

Jae-Woo Chang

Department of Computer Engineering, Chonbuk National University
Chonju, Chonbuk 560-756, South Korea
Phone: +82-652-270-2414 (FAX: 270-2418)
jwchang@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, called 27DLT one, by extending the conventional 9DLT(Direction Lower Triangular) scheme so that we can describe three-dimensional spatial relationships between icon objects accurately. In order to accelerate image searching, we also design an efficient retrieval method using a signature file technique. Finally, we evaluate the retrieval performance of the proposed 27DLT scheme in terms of retrieval effectiveness.

15.1 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. Finally, the representation of original images into iconic images enables users to achieve domain independence and to deal with a group of icon objects in a systematic way.

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 schemes, called 27DLT one, to support the content-based image retrieval in an effective way. The proposed 27DLT scheme can describe three-dimensional spatial relationships between icon objects in a precise way by extending the conventional 9DLT scheme. 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 15.2. The proposed 27DLT representation scheme is described in Section 15.3. A new efficient retrieval method to accelerate image searching is presented in Section 15.4. Section 15.5 provides the performance evaluation of the proposed 27DLT scheme in terms of retrieval effectiveness. Section 15.6 shows user interfaces for iconic image indexing and querying. Finally, Section 15.7 concludes the paper with some issues for future research.

15.2 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 representation schemes using signature file techniques, namely the 2D(Dimensional)-string scheme [2] and the 9DLT(Direction Lower Triangular) scheme [4].

15.2.1 The 2D-string 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. Here, a symbol in the symbolic image corresponds to an object in its original image. In addition, they defined three types (type-0, type-1, and type-2) of 2D sequence pattern matching. For type-0 matching, an arbitrary number of symbols, rows, and columns can be deleted from a symbolic image and can be merged together in order to make it the same as a pattern. Type-1 matching is the same as type-0, except that adjacent rows or columns of a symbolic image cannot be merged. Type-2 matching does not permit any rows and columns to be deleted from a symbolic image.

Lee and Shan [2] proposed a 2D-string representation 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. These signatures are retrieved by either specifying a symbol or specifying a type-i match for i=0, 1, or 2. In addition, they adopted a superimposed coding technique to use the spatial relationships among symbols in a symbolic image as well as to filter quickly for any of the four types of queries. 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:

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

$$V^{AB} = "2" + str(r(A) - r(B)) \quad \text{if } r(A) > r(B)$$

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.

15.2.2 The 9DLT scheme

Chang [3] proposed 9DLT direction codes to describe the type-1 spatial relationship embedded in a 2D string. Therefore, 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 15.1 shows the nine direction codes, where R indicates the reference symbol, 1 stands for "north of R," 2 stands for "northwest of R," and 0 stands for "at the same location as R," and so on.

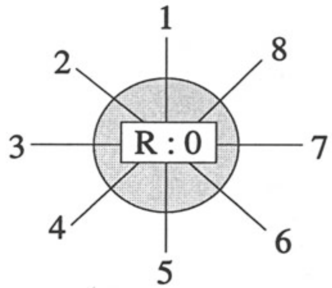


Figure 15.1: 9DLT direction codes

Chang and Jiang [4] proposed a 9DLT representation scheme to express three types of spatial strings 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 scheme describes a spatial representation between A and B symbols as follows.

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

$D'_{AB} = 0$	if $D_{AB} = 0$
$D'_{AB} = 0$ and 1	if $D_{AB} = 1$
$D'_{AB} = 0$ and 3	if $D_{AB} = 3$
$D'_{AB} = 0$ and 5	if $D_{AB} = 5$
$D'_{AB} = 0$ and 7	if $D_{AB} = 7$
$D'_{AB} = 0,1,2$ and 3	if $D_{AB} = 2$

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

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, $|-2| = 2$.

15.3 A NEW THREE-DIMENSIONAL 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 15.2.

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 can represent spatial relationships between icon objects for only the two-dimensional(2D) images. As a result, they are not suitable to expressing spatial relationships between objects for handling three-dimensional(3D) images. In order to support 3D content-based image retrieval, we propose a new three-dimensional representation scheme for iconic image indexing and querying, called 27DLT one, by extending the conventional 9DLT scheme.

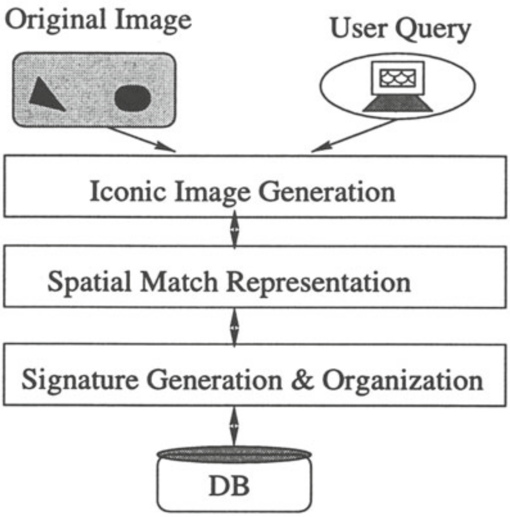


Figure 15.2: Architecture of a spatial match retrieval system

15.3.1 27DLT representation scheme

To describe the spatial relationship between icon objects for 3D images, we propose 27DLT direction codes which extends the 9DLT codes for handing relationship embeded in a 3D images. Therefore, twenty-seven integers from -8 to 18 are used to represent pairwise spatial relationships embeded in a 3D string. Figure 15.3 shows the twenty-seven direction codes, where R indicates the reference icon object. The codes from 1(North) to 8(Northeast) denote directions in a counterclockwise order on the same plane as R and 0 stands for "at the same location on the same plane as R". The codes from -1 to -8 denote the same directions as those of 1 to 8 except that they are described on the inner plane of R. The integer 0 stands for "at the same location as R on the inner plane". Similary, the codes from 11 to 18 denote the same directions as those of 1 to 8 on the outer plane of R. The integer 10 stands for "at the same location as R on the outer plane".

Thus, an exact-match direction character, RE_{AB} is a character describing the 3D spatial relationship between objects A and B when the projects of A and B are represented as a point in a 3D space, respectively. The exact-match direction character is written as the following:

RE_{AB} = 1, -1, 11	if B is north of A in the same, the inner, the outer plane, respectively.
RE_{AB} = 2, -2, 12	if B is northwest of A in the three planes
RE_{AB} = 3, -3, 13	if B is west of A in the three planes
RE_{AB} = 4, -4, 14	if B is southwest of A in the three planes
RE_{AB} = 5, -5, 15	if B is south of A in the three planes
RE_{AB} = 6, -6, 16	if B is southeast of A in the three planes

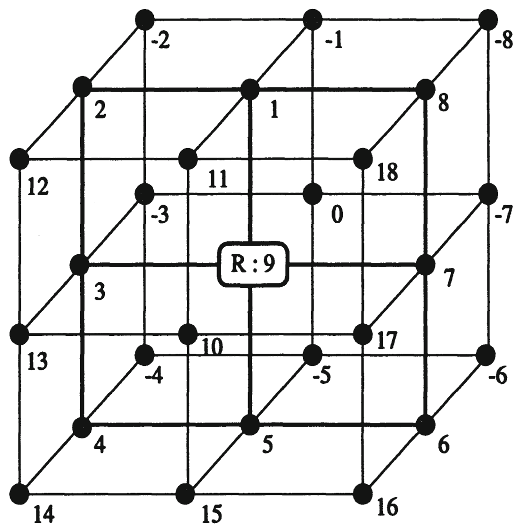


Figure 15.3: 27DLT direction codes

Table 15.1: Approximate-match direction characters

RE_{AB}	RA_{AB}	RE_{AB}	RA_{AB}	RE_{AB}	RA_{AB}
9	9	0	0	10	10
1	1 and 9	-1	-1 and 0	11	10 and 11
3	3 and 9	-3	-3 and 0	13	10 and 13
5	5 and 9	-5	-5 and 0	15	10 and 15
7	7 and 9	-7	-7 and 0	17	10 and 17
2	1, 2, 3, and 9	-2	-1, -2, -3 and 0	12	10, 11, 12 and 13
4	3, 4, 5, and 9	-4	-3, -4, -5 and 0	14	10, 13, 14 and 15
6	5, 6, 7, and 9	-6	-5, -6, -7 and 0	16	10, 15, 16 and 17
8	1, 7, 8, and 9	-8	-1, -7, -8 and 9	18	10, 11, 17 and 18

RE_{AB} = 7, -7, 17   if B is east of A in the three planes
RE_{AB} = 8, -8, 18   if B is northeast of A in the three planes
RE_{AB} = 9, 0, 10   if B is at the same location as A in the three planes

In addition, an approximate-match direction character, RA_{AB} is a character describing the 3D spatial relationship between objects A and B when the projects of A and B are represented as a point in a 3D space, respectively. The approximate-match direction character is written as Table 15.1.

Based on the 27DLT direction codes, we propose a 27DLT representation scheme to express both exact-match and approximate-match of spatial strings so that we can support the full description of pairwise spatial relationships embeded in a 3D string. The exact-match spatial string can be classified into two groups, i.e., general exact-match and distance-considered exact-match, depending on whether or not a distance between two objects is consideted in addition to a direction between them when we represent an iconic image as spatial strings. For expressing the distance, two distance degrees are used; N(near) and F(far-away). The optimal threshold value of differenciating between 'N' and 'F' can be determined by a large number of experiments, but a naive value can be determined to be the average distance between two objects. The three-dimensional distance character between objects A and B over X-,Y-, and Z-axis, D_{AB}, is shown in Table 15.2.

Table 15.2: Distance character between objects A and B

distance character	distance over X-axis	distance over Y-axis	distance over Z-axis
0	N	N	N
1	N	N	F
2	N	F	N
3	N	F	F
4	F	N	N
5	F	N	F
6	F	F	N
7	F	F	F

An approximate-match spatial string of objects A and B, STA_{AB}, a general exact-match spatial string, STE_{AB}, and a distance-considered exact-match spatial string, STD_{AB}, are expressed as follows:

- approximate-match string
  $SBA^{AB} = \{(A, B, RA_{AB}\}$
- general exact-match string
  $STE^{AB} = \{(A, B, RE_{AB}\}$
- distance-considered exact-match string
  $STD^{AB} = \{(A, B, RE_{AB}, D_{AB}\}$

15.4 AN 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 technique because of its main advantages: fast retrieval time and low storage overhead [6, 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 addition, we construct a general exact-match signature by superimposing all of the signatures, each of which is made from each general exact-match spatial string for the iconic image. Similarly, we create a distance-considered exact-match signature for the iconic image. Superimposing signatures leads to reducing the disk space to be accessed dramatically. Using a disjoint coding technique, we finally construct an image signature by concatenating the object signature, the approximate-match one, and the superimposing one of both the general exact-match and the distance-considered exact-match signature.

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.

15.4.1 Signature generation

With a set of distance-considered exact-match spatial relationship strings (DESRs) corresponding to a given iconic image, we can generate a set of general exact-match spatial relationship strings (GESRs), approximate-match spatial relationship strings (ASRs), and an object list (OL). Given the OL, a set of ASRs, a set of GESRs, and a set of DESRs, we can also generate four kinds of signatures for the iconic image, i.e., object, approximate-match, general exact-match, and distance-considered exact-match ones. Then, an image signature for the iconic image is constructed by concatenating the object one, the approximate-match one, and the superimposing one of both general and distance-considered exact-match signatures. The algorithm to generate an image signature is illustrated below.

[Algorithm 1] Generation of image signature
Input: a set of DESRs for an iconic image, each being (A,B,R_{AB},D_{AB})
Output: image signature, IS
Variables:
S_{obj}, S_{app}, S_{gex}, S_{dex} : object, approximate-match, and general exact -match, distance-considered exact-match signature for an iconic image, respectively
so_k : object signature for the k-th object of the OL
sa_i, sge_i, sde_i, : approximate-match, general exact-match, and distance-considered exact-match signature for the i-th DESR, respectively
s_i^j : approximate-match signature for the j-th ASR of the i-th DESR
Begin:
$S_{obj} = 0$; $S_{app} = 0$; $S_{gex} = 0$; $S_{dex} = 0$;
Compute the OL from a set of DESRs;

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 DESR for some i) {
  Create sde_i from the i-th DESR; $S_{dex} = S_{dex} \vee sde_i$;
  Determine a GESR from the i-th DESR;
  Create sge_i from the GESR; $S_{gex} = S_{gex} \vee sge_i$;
  Determine a set of ASRs from the i-th DESR; $sa_i = 0$;
  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 */
$RS = S_{obj} || S_{app} || (S_{gex} \vee S_{dex})$;

End:

15.4.2 Insertion and Retrieval

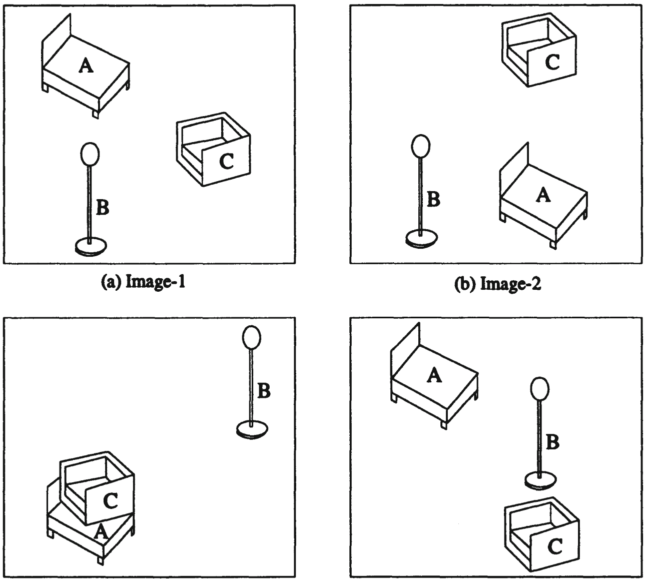
When a set of signatures for an iconic image is generated using Algorithm 1, we can store the object signature and the approximate-match signature into an object signature file and an approximate-match one, respectively. We also store the superimposing one of both general and distance-considered signatures into an exact-match signature file. 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.

15.4.3 Example

We assume that we have four iconic images consisting of icon objects A, B, and C as shown in Figure 15.4. A set of approximate-match, general exact-match, and distance-considered exact-match spatial relationship strings in our basic representation can be obtained as follows:

- approximate-match representation
  (Image-1) (A,B,10),(A,C,10),(A,C,17),(B,C,-7),(B,C,0)
  (Image-2) (A,B,3),(A,B,9),(A,C,0),(B,C,-7),(B,C,0)



(a) Image-1 (b) Image-2 (c) Image-3 (d) Image-4

Figure 15.4: Four example iconic images

(Image-3) (A,B,-7),(A,B,0),(A,C,1),(A,C,9),(B,C,10),(B,C,13)
(Image-4) (A,B,10),(A,B,17),(A,C,10),(A,C,17),(B,C,10)

- general exact-match representation
  (Image-1) (A,B,10),(A,C,17),(B,C,-7)
  (Image-2) (A,B,3),(A,C,0),(B,C,-7)
  (Image-3) (A,B,-7),(A,C,1),(B,C,13)
  (Image-4) (A,B,17),(A,C,17),(B,C,10)
- distance-considered exact-match representation
  (Image-1) (A,B,10,1),(A,C,17,0),(B,C,-7,0)
  (Image-2) (A,B,3,0),(A,C,0,1),(B,C,-7,1)
  (Image-3) (A,B,-7,5),(A,C,1,0),(B,C,13,5)
  (Image-4) (A,B,17,0),(A,C,17,5),(B,C,10,0)

To create image signatures for the four iconic images, we assume that an object signature has 8 bits in length, an approximate-match signature has 16 bits, and each exact-match signature has 16 bits. In addition, we assume that four hashing functions are used to generate these signatures. Table 15.3, Table 15.4, Table 15.5, and Table 15.6 list the object, the approximate-match, the general exact-match, and the distance-considered exact-match signatures, respectively. Based on them, we can generate image signatures for the four iconic images as shown in Table 15.7.

Figure 15.6 illustrates a signature file structure after we insert the four image signatures in Table 15.7. Here the SRS file is the one storing a set of DESRs,

Table 15.3: Object signatures

object	object signature
A	00010001
B	00100010
C	01000100

Table 15.4: Approximate-match signatures

ASR	approximate-match signature
(A,B,-7)	00000000 00000001
(A,B,0)	00000000 00000010
(A,B,3)	00000000 00000100
(A,B,9)	00000000 00001000
(A,B,10)	00000000 00010000
(A,B,17)	00000000 00100000
(A,C,0)	00000000 01000000
(A,C,1)	00000000 10000000
(A,C,9)	00000001 00000000
(A,C,10)	00000010 00000000
(A,C,17)	00000100 00000000
(B,C,-7)	00001000 00000000
(B,C,0)	00010000 00000000
(B,C,10)	00100000 00000000
(B,C,13)	01000000 00000000

Table 15.5: General exact-match signatures

GESR	general exact-match signature
(A,B,-7)	00000000 00000001
(A,B,3)	00000000 00000010
(A,B,10)	00000000 00000100
(A,B,17)	00000000 00001000
(A,C,0)	00000000 00010000
(A,C,1)	00000000 00100000
(A,C,17)	00000000 01000010
(B,C,-7)	00000000 10000100
(B,C,10)	00000001 00001000
(B,C,13)	00000010 00010000

GESRs, and ARSs. For example, suppose that we have a query to find such an iconic image as Image-Q in Figure 15.5. To answer this query, we first generate a set of SRS for Image-Q in our basic representation as follows:

Table 15.6: Distance-considered exact-match signatures

DESR	distance-considered exact-match signature
(A,B,-7,5)	10000000 00000001
(A,B,3,0)	01000000 00000010
(A,B,10,1)	00100000 00000100
(A,B,17,0)	00010000 00001000
(A,C,0,1)	00001000 00010000
(A,C,1,0)	00000100 00100000
(A,C,17,0)	00000010 01000010
(A,C,17,5)	00000001 01000010
(B,C,-7,0)	00000000 10000100
(B,C,-7,1)	00000000 01000100
(B,C,10,0)	00000000 00100000
(B,C,10,1)	00000000 00010000
(B,C,13,5)	00000000 00001000

Table 15.7: Image signatures for the four example iconic images

Image	IS_{obj}	IS_{app}	$IS_{gex} \vee IS_{dex}$
Image-1	01110111	00011110 00010000	00100010 11000100
Image-2	01110111	00011000 01001100	01001000 11010010
Image-3	01110111	01100001 10000011	10000110 00101001
Image-4	01110111	00100110 00110000	00010001 01101000

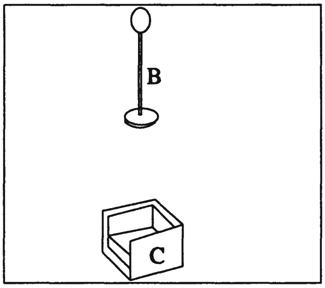


Figure 15.5: Image-Q: A query iconic image

- approximate-match representation
  (B,C,10)
- general exact-match representation
  (B,C,10)
- distance-considered exact-match representation
  (B,C,10,1)

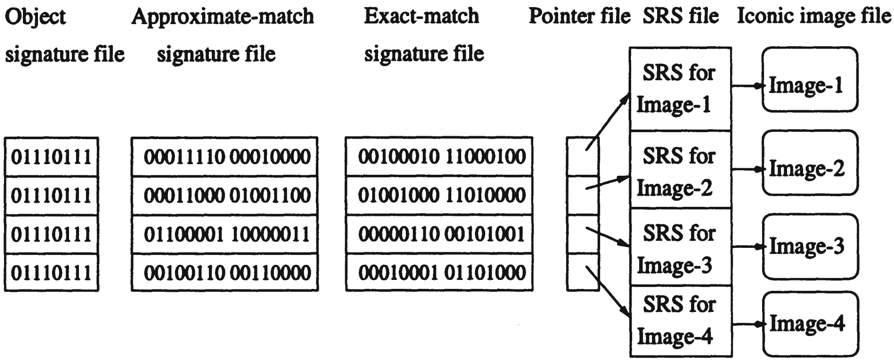


Figure 15.6: New signature file organization

Next, we create the object, the approximate-match, the general exact-match, and the distance-considered exact-match signatures for Image-Q by using Table 15.3, Table 15.4, Table 15.5, and Table 15.6 as follows:

$$IS^Q_{obj} = 01100110$$
$$IS^Q_{app} = 00100000\ 00000000$$
$$IS^Q_{gex} = 00000001\ 00000000$$
$$IS^Q_{dex} = 00000000\ 00010000$$

If we require some distance-considered exact-match answers, we can compare IS^Q_{dex} with the four signatures in the exact-match signature file. we can obtain one qualifying signature because the second signature in the exact-match signature file satisfies the bit pattern of IS^Q_{dex}. However, it is proved to be a false match because the corresponding DESRs for image-2 does not actually contain the DESR of the query. On the other hand, if we require some approximate-match answers, we can search for only the four signatures of the approximate-match signature file. We obtain two qualifying signatures, i.e., the third and the fourth of the approximate-match signature file, because they contain the bit pattern of IS^Q_{app}. Thus, we can access the ARS 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 two qualifying iconic images, i.e., Image-3 and Image-4, because both the ARSs of Image-3 and those of Image-4 both include the ARS of Image-Q.

15.5 PERFORMANCE EVALUATION

We assume that an iconic image consists of icon objects, each having its icon name and its position. For our 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 is 10,000.
- A query iconic image contains two to five icon objects.

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:

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

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

When 1000 different queries are executed, Table 15.8 shows the retrieval effectiveness of our 27DLT representation scheme, in terms of precision measure. Here, "Exact" means a query type for the general exact match and "Approx." means one for the approximate match. Our 27DLT representation scheme achieves nearly the same retrieval precision, compared to the 9DLT scheme. 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.

Table 15.8: Precision measure of our 27DLT scheme

# of icon objects	9DLT Scheme		Our 27DLT scheme	
in a query	Approx.	Exact	Approx.	Exact
2	0.12	0.19	0.13	0.18
3	0.39	0.52	0.41	0.48
4	0.45	0.50	0.50	0.54
5	0.47	0.53	0.52	0.53
Precision	0.35	0.43	0.39	0.43

In order to verify the correctness of our experiment, we also implemented our 27DLT 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 ten queries are executed, Table 15.9 shows the retrieval effectiveness of our 27DLT representation scheme, in terms of precision and recall measures. Our 27DLT representation scheme shows approximately the same retrieval effectiveness results, compared to those in Table 15.8. In case of the approximate-match query, the precision values of our implementation with real images are a little lower than those of our experiment with imaginary images. This is because we don't have a large number of real images enough to obtain sufficient qualifying images to answer a given query. It is shown from the results that our 27DLT representation scheme holds about 0.3 precision value in the approximate match and about 0.5 in the exact match, while their recall values are kept one.

Table 15.9: Retrieval effectiveness of our 27DLT scheme

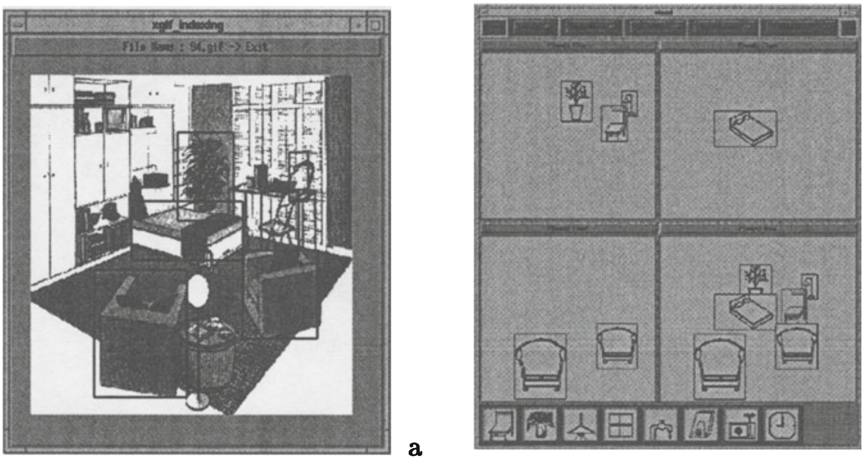
retrieval	9DLT Scheme		Our 27DLT Scheme	
effectiveness	Approx.	Exact	Approx.	Exact
Precision	0.2	0.38	0.22	0.44
Recall	1.0	1.0	1.0	1.0

15.6 USER INTERFACES FOR ICONIC IMAGE GENERATION

As shown in Figure 15.2, some original images are transformed into their iconic images through the iconic image generation step, prior to their storage to the database. In addition, a user query can be transformed into its iconic image query through this step so that we search for qualifying iconic images to satisfy the query. In order to make this step easy-to-use, we implemented user interfaces for both image indexing and querying.

15.6.1 User interface for image indexing

For image indexing of a pixel-level original image, a large number of known image processing and understanding techniques [5] can be used to identify some domain objects and their positions in the original image. For example, the MBR(Minimum Bounding Rectangle) technique can represent the positional relationships among domain objects in a precise manner because it can preserve the size of the domain objects. Though the image processing and understanding task is computationally expensive and difficult, it is performed only at the time of image insertion into the database. We provide a user interface where a pixel-level original image can be transformed into an iconic image by manually associating each domain object in an original image with a meaningful icon object. Figure 15.7 shows the user interface for image indexing. A set of domain objects surrounded by the six rectangles of the original image in Figure 15.7(a) can be transformed into their corresponding iconic objects by using this interface. In Figure 15.7(b), the iconic objects of the upper left, the upper right, and the bottom left indicate those appeared in the inner plane, the same plane, and the outer plane, respectively. Therefore, an iconic image of the bottom right in Figure 15.7(b) is made by combining all the iconic objects in the three planes.



a b

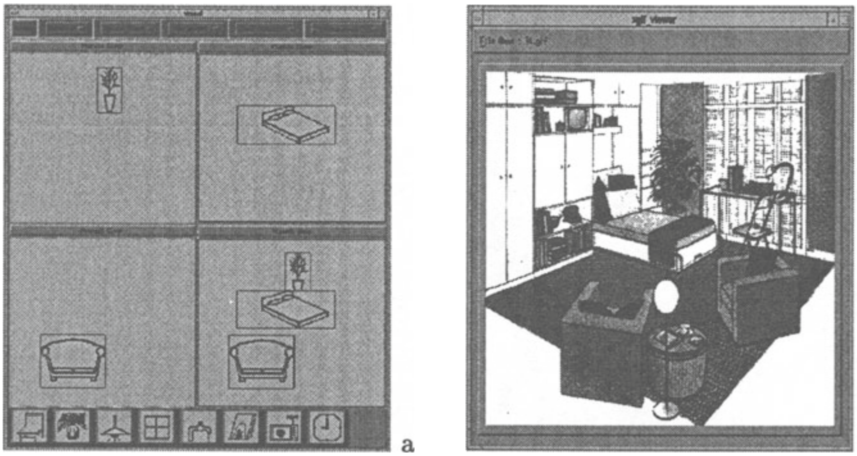
Figure 15.7: Image indexing interface

15.6.2 *User interface for image querying*

For image retrieval, a user query can first be transformed into an iconic image in the same way as the image processing and understanding techniques. The traditional user interface is not appropriate for users to retrieve relevant images in a convenient way. Thus, it is necessary to make a visual user interface where users can easily construct their query being expressed as icon objects. Therefore we impliment a query-by-iconic image interface in order to provide users with a more convenient tool for writing their query. In Figure 15.8(a), the query consists of three iconic objects each of which comes from the inner, the same, and the outer plane, respectively. As shown in Figure 15.8(b), we obtain only one qualifying image to satisfy the query since we use a small database with 100 real interior design images.

15.7 CONCLUSIONS AND FUTURE WORK

Recently, much attention has been paid to Multimedia Information Retrieval (MIR) because there are so many applications which require multimedia data. In order to support MIR in an effective way, content-based image retrieval is essential for retrieving relevant multimedia documents. For this, we proposed our 27DLT spatial match representation scheme so as to support three-dimensional content-based image retrieval in an effective way. Our representation scheme accurately described three-dimensional spatial relationships between icon ob-



a b

Figure 15.8: Image querying interface

jects because they could make use of 27DLT direction codes. To accelerate searching, we also designed our efficient retrieval method based on a signature file technique.

In order to prove the superiority of our 27DLT scheme on retrieval effectiveness, we evaluated its retrieval performance in terms of both precision and recall measures. We showed from our experiment that our 27DLT scheme holds about 0.3 precision value in the approximate match and about 0.5 in the exact match, while its recall values are kept one. As further work, our 27DLT representation scheme should be applied to real application areas using three-dimensional 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] 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.

[7] J. W. Chang, J. H. Lee, Y. J. LEE. Multikey Access Methods Based on Term Discrimination and Signature Clustering. In *ACM 12th International Conference on Research and Development in Information Retrieval*, pages 176-185, Cambridge, Massachusetts, Jun, 1989.

[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.