아이콘 이미지 데이타베이스를 위한 시그니쳐에 기반한 공간-매치 검색 기법

장 재 우[†] · Jaideep Srivastava[††]

요 약

멀티미디어 정보 검색 응용분야에서 내용-본위 이미지 검색은 유사성이 높은 멀티미디어 문서를 검색하는데 있어서 필수적이다. 따라서, 본 논문은 픽셀(pixel) 단위의 본래 이미지가 자동적 혹은 수동적으로 아이콘(icon) 객체로 구성된 아이콘 이미지로 변환될 때, 아이콘 이미지의 효율적인 표현과 검색을 제공하기 위함이다. 이를 위해 먼저 아이콘 객체를 점 대신에 사각형으로 표현함으로써 아이콘들간의 공간관계를 정확하게 표현할 수 있는 새로운 공간 매치 표현 기법을 제안한다. 또한, 이미지의 검색을 가속화시키기 위해, 2차원 시그니쳐 화일 구성을 사용한 효율적인 검색기법을 설계한다. 마지막으로, 성능 실험을 통하여 제안한 공간 매치 표현 기법이 기존의 9-DLT 표현 기법보다 더 나은 검색 효율을 나타냄을 보인다.

A Signature-based Spatial Match Retrieval Method for Iconic Image Databases

Jae-Woo Chang[†] · Jaideep Srivastava[††]

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 effective representation and efficient retrieval of images when a pixel-level original image is automatically or manually transformaed into its iconic image containing meaningful graphic descriptions, called icon objects. For this, we first propose new spatial match representation schemes to describe spatial relationships between icon objects accurately by expressing them as rectangles, rather than as points. In order to accelerate image searching, we also design an efficient retrieval method using a two-dimensional signature file organization. Finally, we show from our experiment that the proposed representation schemes achieve better retrieval effectiveness than the 9-DLT(Direction Lower Triangular) 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 handing multimedia data, such as text, image, video, and animation. The applications include digital libraries, advertisements, medical information, remote sensing and astronomy, cartography, digital newspapers,

※본 연구는 한국과학재단 95년도 후반기 해외 post-doc 연수에 의해 수행되었음
† 종신회원:전북대학교 컴퓨터공학과
†† 정 회 원:Dept. of Computer Sciences, University of Minnesota, USA
논문접수:1997년 4월 9일, 심사완료:1997년 10월 18일

and architectural design. So far, attributes in multimedia dcuments 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 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 user quaeries 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 transformaed into its iconic image including icon objects. For this, we propose new spatial match representation schemes to support the content-based image retrieval in an effective way. The proposed representation schemes can describe spatial relationships between icon objects in a precise way because they represent the icon objects as rectangles rather than as points, and they make use of accurate positional operators. In order to accelerate image searching, we also design an efficient retrieval method using a two-dimensional signature file organization.

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 schemes are described in Section 3. An efficient retrieval method to accelerate image searching is presented in Section 4. Section 5 provides a comparison of the proposed representation schemes with the conventional 9-DLT one in order to prove the superiority of our schemes. 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-string based scheme[2] and the 9-DLT 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. 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 mathcing, 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 mathcing does not permit any rows and columns to be deleted from a symbolic image.

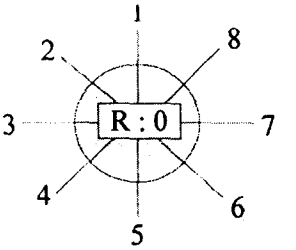
Lee and Shan[2] proposed a 2D-string based scheme to express some types of spatial relationships of symbolic images. In this scheme, they generated four kinds of two-level signature files by associating each symbolic image with a record signature and by relating some images with a block signature. These 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:

(type-0) V^{AB} = "0" if r(A) = r(B)
V^{AB} = "0" and "1" if r(A) < r(B)
V^{AB} = "0" and "2" if r(A) > r(B)
(type-1) V^{AB} = "0" if r(A) = r(B)
V^{AB} = "1" if r(A) < r(B)
V^{AB} = "2" if r(A) > r(B)
(type-2) V^{AB} = "0" + str(r(A) − r(B)) if r(A) = r(B)
V^{AB} = "1" + str(r(B) − r(A)) if r(A) < r(B)
V^{AB} = "2" + str(r(B) − r(A)) 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, tr(3) = "3". A type-*i* 2D spatial string for symbols A and B when i = 0, 1, and 2, S^{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. For example, when a symbolic projection is given as shown in Fig 1, a set of type-0, type-1, and type-2 representation strings for the example can be obtained as follows:

type-0 : (A, B, 2, 1), (A, B, 2, 0), (A, B, 0, 1), (A, B, 0, 0), (A, C, 1, 0), (A, C, 0, 0), (B, C, 1, 2), (B, C, 1, 0), (B, C, 0, 2), (B, C, 0, 0)
type-1 : (A, B, 2, 1), (A, C, 1, 0), (B, C, 1, 2)
type-2 : (A, B, 2, 1, 1, 1), (A, C, 1, 1, 0, 0), (B, C, 1, 2, 2, 1)

(Fig. 1) Example symbolic image



(Fig. 2) 9-DLT direction codes

2.2 The 9-DLT based scheme

Chang[3] proposed a 9-DLT representation to describe the type-i 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. Fig 2 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.

Chang and Jiang[4] proposed a 9-DLT based scheme to express three types of spatial strings by extending the 9-DLT 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 9-DLT based 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 $D_{AB} = 2$

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

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

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

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

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

$SC_X^{AB} = 0$ if $|rx(A) - rx(B)| \leq 1$

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

$SC_Y^{AB} = 0$ if $|rx(A) - rx(B)| \leq 1$

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

Here, S_i^{AB} represents the type-i spatial strings for A and B symbols, and (A, B, D_{AB}) denotes the 9-DLT 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, $|-2| = 2$. For example, when we have the same symbolic projection as that in Fig 1, a set of type-0, type-1, and type-2 representation strings can be obtained as follows:

type-0: (A, B, 0), (A, B, 1), (A, B, 2), (A, B, 3), (A, C, 0), (A, C, 7), (B, C, 0), (B, C, 5), (B, C, 6), (B, C, 7)

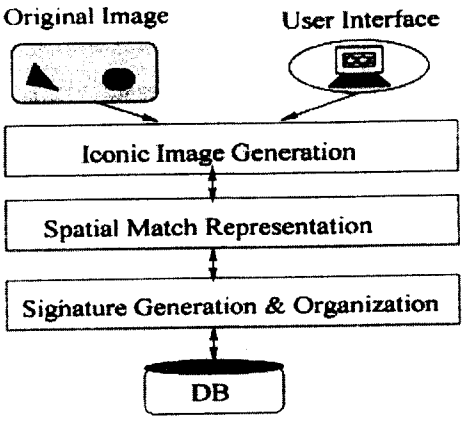
type-1: (A, B, 2), (A, C, 7), (B, C, 6)

type-2: (A, B, 2, 0, 0), (A, C, 7, 0, 0), (B, C, 6, 1, 0)

3. New Spatial Match Representation Schemes

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, an inconic image can be easily obtained by associating a meaningful icon object with each domian 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, the query iconic image can be represented as spatial strings by using some spatial match representations. Then, a query signature can be generated from the spatial strings and some potential matches can be obtained after the query signature compares with all of the signatures in the signature file. Finally some iconic images satisfying the user query can be retrieved when some false matches are excluded from the potential ones. The architecture of a spatial match retrieval system is shown in Fig 3.

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

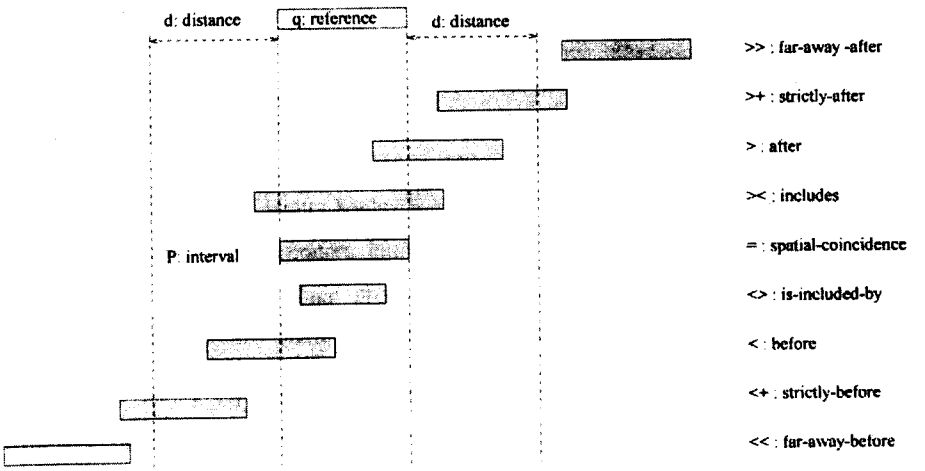


(Fig. 3) The architecture of a spatial match retrieval system

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 new spatial match representation schemes to support effective content-based image retrieval. The proposed representation schemes, i.e., basic and detailed ones, can accurately describe spatial relationships between icon objects of iconic images because they represent each icon object as a rectangle, rather than as a point.

3.1 Basic 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 positional operator, which relates the intervals originated by the projections of p and q on the X-axis (or Y-axis). For this, we propose nine positional operators which can express basic ones among all of the possible relationships between a pair intervals. The proposed nine operators are easy-to-handle and also sufficient to handle such images that their objects rarely have an adjacency with the other objects. These images are widely used in many application dealing with real photos like human face images. In the case of the X axis, 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 the distance between two objects. The naive value of d can be determined as the average



(Fig. 4) Positional operators of our basic representation scheme.

length of reference intervals, and the optimal value can be determined by a large number of experiments.

(a) p far-away-after($\gg$)q iff $x_{p1} \geq x_{q2} + d$
(b) p strictly-after($>+$)q iff $x_{q2} < x_{p1} < x_{q2} + d$
(c) p after($>$)q iff $x_{q1} < x_{p1} \leq x_{q2}$ and $x_{q2} < x_{p2}$
(d) p includes ($><$)q iff($x_{p1} \leq x_{q1}$ and $x_{q2} < x_{p2}$) or ($x_{p1} < x_{q1}$ and $x_{q2} \leq x_{q2}$)
(e) p spatial-coincidence ($=$)q iff $x_{p1} = x_{q1}$ and $x_{p2} = x_{q2}$
(f) p is-included-by ($<>$)q iff($x_{q1} \leq x_{p1}$ and $x_{p2} < x_{q2}$) or ($x_{q1} < x_{p1}$ and $x_{p2} \leq x_{q2}$)
(g) p before ($<$)q iff $x_{q1} \leq x_{p2} < x_{q2}$ and $x_{p1} < x_{q1}$
(h) p strictly-before ($<+$)q iff $x_{p2} < x_{q1} < x_{p2} + d$
(i) p far-away-before($\ll$)q iff $x_{q1} \geq x_{p2}$

The visual sketch of the meanings of positional operators used for our basic representation scheme is given in Fig 4. For 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 fully support the description of all the spatial relationships embedded in a 2D string[2]. Thus, the type-0, 1, and 2 spatial relationships encoded by a 2D string can be transformed into exact-match and approximate-match spatial strings for our basic representation scheme. An exact-match i-axis spatial character, BE_i^{AB}, is a character describing the spatial relationship between A and B objects 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 $x_{q1} < x_{q2}$. This character is used to support the exact match of user queries holding certain spatial relationships. The exact-match spatial chatial character is written as the following:

$BE_i^{AB} = 0$ if $p \gg q$ $BE_i^{AB} = 5$ if $p <> q$
$BE_i^{AB} = 1$ if $p >+ q$ $BE_i^{AB} = 6$ if $p <> q$
$BE_i^{AB} = 2$ if $p > q$ $BE_i^{AB} = 7$ if $p <+ q$
$BE_i^{AB} = 3$ if $p >< q$ $BE_i^{AB} = 8$ if $p \ll q$
$BE_i^{AB} = 4$ if $p = q$

An approximate-match i-axis spatial character, BA_i^{AB}, is a character describing the spatial relationship between objects A and B, being used to support the appoximate match of user queries. To determine if positional operator holds a approximate-match relationship with another operator, we classify nine positional operators into five groups, i.e., *after* group($\gg$, $>+$, $>$), *is-included* group($<>$), *same* group($=$), *include* group($><$), and *before* group ($<$, $<+$, $\ll$). A procedure to determine approximate-match relationships among operators are as follows:

1. Determine intra-group approximate-match relationships among the three operators for the *after* group as well as for the *before group*, respectively.
2. Establish no approximate-match relationship between two operators, one coming from the *after* group and the other coming from the *before* group. This is because there is no approximate-match relationship between the two groups.
3. Similarly, establish no approximate-match relationship between two operators, one coming from the *is-included* group and the other from the *include* group.
4. Finally, establish approximate-match relationships between the *same* group and the *include* group, as well as between the *same* group and the *is-include d* group.

According to this procedure, the approximate-match character is written as the following:

$BA_i^{AB} = 0$ and 1 if $p \gg q$ $BA_i^{AB} = 5$ and 6 if $p = q$
$BA_i^{AB} = 0$, 1 and 2 if $p >+ q$ $BA_i^{AB} = 6$ and 7 if $p <> q$
$BA_i^{AB} = 1$ and 2 if $p > q$ $BA_i^{AB} = 6$, 7 and 8 if $p <+ q$
$BA_i^{AB} = 3$ and 4 if $p >< q$ $BA_i^{AB} = 7$ and 8 if $p \ll q$

Therefore, an exact-match spatial string of objects

A and B, SBE^{AB}, is a string formed by concatenating A, B, and exact-match spatial characters BE_X^{AB} and BE_Y^{AB}, where BE_X^{AB} is the spatial character along the X-axis, and BE_Y^{AB} is the spatial character along the Y-axis. Similarly, an approximate-match spatial string of objects A and B, SBA^{AB}, is a string formed by concatenating A, B, and approximate-match spatial characters BA_X^{AB} and BA_Y^{AB}. Thus, the exact-match and the approximate-match spatial strings of objects A and B are expressed as follows:

- exact-match string
  $SBE^{AB} = \{(A, B, BE_X^{AB}, BE_Y^{AB})\}$
- approximate-match string
  $SBA^{AB} = \{(A, B, BA_X^{AB}, BA_Y^{AB})\}$

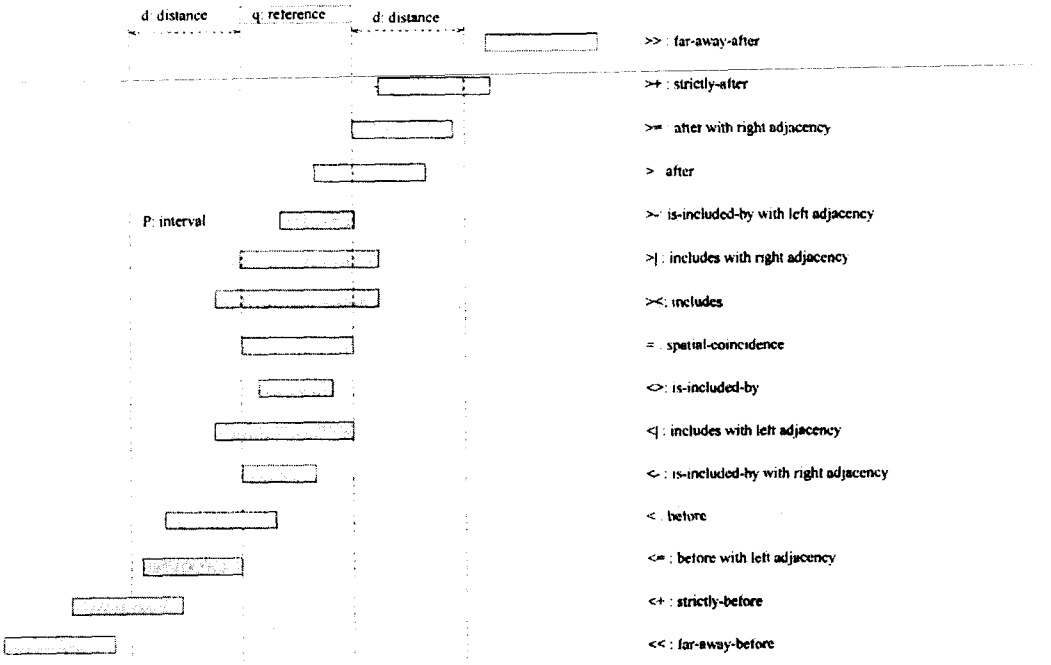
3.2 Detailed representation scheme

For new positional operators, we can extend some operators used for the specification of temporal relationships between time intervalus in interval logic [6, 7]. 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.

(a) p far-away-after($\gg$)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($\Leftarrow$)q
  iff $x_{p2} = x_{q1}$
(n) p strictly-before($<+$)q
  iff $x_{p2} < x_{q1} < x_{p2} + d$
(o) p far-away-before($\ll$)q
  iff $x_{q1} \geq x_{q2} + d$

The visual sketch of positional operators used for our detailed representation scheme are given in Fig 5. 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:

$DE_i^{AB} = 0$ if $p \gg q$ $DE_i^{AB} = 8$ if $p <> q$
$DE_i^{AB} = 1$ if $p \gg +q$ $DE_i^{AB} = 9$ if $p < | q$
$DE_i^{AB} = 2$ if $p >= q$ $DE_i^{AB} = 10$ if $p < - q$
$DE_i^{AB} = 3$ if $p > q$ $DE_i^{AB} = 11$ if $p < q$
$DE_i^{AB} = 4$ if $p > - q$ $DE_i^{AB} = 12$ if $p <= q$
$DE_i^{AB} = 5$ if $p > | q$ $DE_i^{AB} = 13$ if $p < + q$
$DE_i^{AB} = 6$ if $p >< q$ $DE_i^{AB} = 14$ if $p \ll q$
$DE_i^{AB} = 7$ if $p = q$



(Fig. 5) Positional operations of our detailed representation scheme

An approximate-match i-axis spatial character, DA_i^{AB}, is a character describing a spatial relationship between A and B objects so that it can be used to support the approximate match of user queries. To determine if a positional operator holds an approximate-match relationship with another operator, we classify the fifteen positional operators into five groups, i.e., *after* group($\gg$, $>$, $>+$, $>=$, $>$), *is-included* group($>-$, $<>$, $<-$), *same* group($=$), *includ* group($>|$, $><$, $<|$), and *before* group($<$, $<=$, $<+$, $\ll$). A procedure to determine approximate-match relationships among operators are as follows:

1. Determine intra-group approximate-match relationships among all operators for each group.
2. Establish no approximate-match relationship between two operators, on belonging to the *after* group and the other belonging to the *before* group. This is because there is no approximate-match relationship between the two groups.
3. Similarly, establish no approximate-match relationship between two operators, one belonging to the *is-included* group and the other belonging to the *include* group.
4. Determine approximate-match relationships between the *after* group and the *is-included* group, as well as between the *after* group and the *include* group. For this, establish an approximate-match relationship between two operators, one coming from the *after* group and the other from the *is-included* group. Also, establish an approximate-match relationship between two operators, one from the *after* group and the other from the *include* group.
5. Similarly, determine approximate-match relationships between the *before* group and the *is-included* group, as well as between the *before* group and

the *include* group. For this, establish an approximate-match relationship between two operators, one coming from the *before* group and the other from the *is-included* group. Also, establish are relationship between two operators, one from the *before* group and the other from the *include* group.

6. Finally, determine approximate-match relationship between the *same* group and the *is-included* group, as well as between the *same* group and the *include* group. For this, establish the approximate-match relationships of the *same* operator with all of the operators in the *include* group, as well as with all of the operatiors in the *is-included* group.

According to the above procedure, the approximate-match spatial chareacter is written as the following:

DE_i^{AB} = 0 and 1 if $p \gg q$
DA_i^{AB} = 0, 1 and 2 if $p > +q$
DA_i^{AB} = 1, 2 and 3 if $p >= q$
DA_i^{AB} = 2, 3, 4 and 5 if $p > q$
DA_i^{AB} = 3, 4, 7 and 8 if $p > -q$
DA_i^{AB} = 3, 5, 6 and 7 if $p > |q$
DA_i^{AB} = 5, 6 and 9 if $p >< q$
DA_i^{AB} = 4, 5, 7, 9 and 10 if $p = q$
DA_i^{AB} = 4, 8 and 10 if $p <> q$
DA_i^{AB} = 6, 7, 9 and 11 if $p < |q$
DA_i^{AB} = 7, 8, 10 and 11 if $p < -q$
DA_i^{AB} = 9, 10, 11 and 12 if $p < q$
DA_i^{AB} = 11, 12 and 13 if $p <= q$
DA_i^{AB} = 12, 13 and 14 if $p < +q$
DA_i^{AB} = 13 and 14 if $p \ll q$

Therefore, an exact-match spatial string of objects A and B, SDE^{AB}, is a string formed by concatenating A, B, and exact-match spatial characters DE_X^{AB} and DE_Y^{AB}, where DE_X^{AB} is the spatial character along the X-axis, and DE_Y^{AB} is the spatial character along the Y-axis. Similarly, an approximate-match spatial string of objects A and B, SDA^{AB}, is a string formed by concatenating A, B, and exact-match spatial characters DA_X^{AB} and DA_Y^{AB}. Thus, the exact-match and the approximate-match strings of objects A and B are expressed as follows:

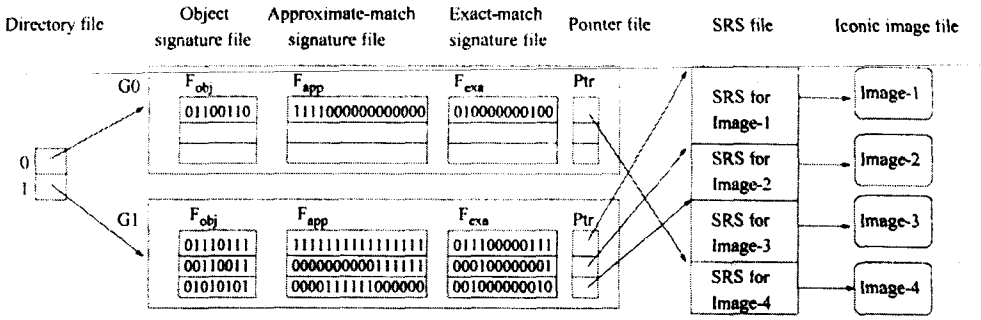
- exact-match string
  $SDE^{AB} = \{(A, B, DE_X^{AB}, DE_Y^{AB})\}$
- approximate-match string
  $SDA^{AB} = \{(A, B, DA_X^{AB}, DA_Y^{AB})\}$

4. An Efficient Retrieval Method

4.1 A signature file organization

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 overthead[8]. 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 objects signatures, thus dramatically reducing the query processing time. Similarly, if a user query requires all relevant images satisfying a certain relatinship approximately, we can access only a portion of



(Fig. 6) A new signature file organization

approximate-match signatures to answer the query. In order to accelerate the search for these signatures, we design an efficient signature file organization based on a two-dimensioanl signature file (TDSF) technique [9] as shown in Fig 6. Here F_{obj}, F_{app}, and F_{exa} indicate a frame block of the object, the approximate-match, the exact-match signature file, respectively. The group G0 indicates a group of frame blocks being number 'zero' in F_{obj}, F_{app}, and F_{exa}. The SRS file is the one storing a set of approximate-match and exact-match spatial relationship strings(SRS) for all of the iconic images.

4.2 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 signautre is illustrated below.

[*Algorithm 1*] Generation of image signautre

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 as set of ESRs;

while(eack 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 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;

4.3 Insertion and Retrieval

When an image signautre is generated using Algorithm 1, we can compute from it the address of an entry to be accessed for the directory file. After searching the entry, we obtain the the addresses of frame blocks, being indicated by a group G, for F_{obj}, F_{app}, and F_{exa}, respectively. In the group G, we finally store the object, the approximate-match, and the exact-match part of the image signature into F_{obj}, F_{app} and F_{exa}, respectively. If the insertion of the image signature causes the overflow of frame blocks, we resolve the overflow by splitting the frame block into two blocks.

When a user query is given, it can be transformed into a query signature using Algorithm 1. After we take a hashing key from the query signature, we can compute a set of equivalent keys based on the key. After searching the descriptor file of the TDSF, we can determine what frame block should be accessed first. Depending on whether a query is an approximate match or an exact match, we can compute a searching order to decide in what sequence three signature files should be accessed so that we may achieve good retrieval performance. This generally depends both on its query type and on the number of 1's in each frame area of a query signature. After searching the corresponding frame blocks in the searching order, we can obtain some qualifying signatures. Finally, we can find iconic image results by examining whether the iconic images corresponding to the qualifying signatures actually satisfy the user 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 to lack of space.

4.4 Example

We assume that we have four iconic images consisting of icon objects A, B, and C as shown in Fig 7. A set of approximate-match and exact-match spatial relationship strings in our basic representation can be obtained as follows:

- *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, C, 6, 5), (A, C, 6, 6), (A, C, 7, 5), (A, C, 7, 6), (A, C, 8, 5), (A, C, 8, 6), (B, C, 7, 6), (B, C, 1, 7), (B, C, 8, 6), (B, C, 8, 7)

(Image-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)

(Image-3) (A, C, 6, 5), (A, C, 6, 6), (A, C, 7, 5), (A, C, 7, 6), (A, C, 8, 5), (A, C, 8, 6)

(Image-4) (B, C, 7, 6), (B, C, 1, 7), (B, C, 8, 6), (B, C, 8, 7)

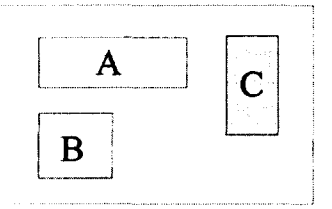
- exact-match representation

(Image-1) (A, B, 5, 1), (A, C, 7, 5), (B, C, 8, 6)

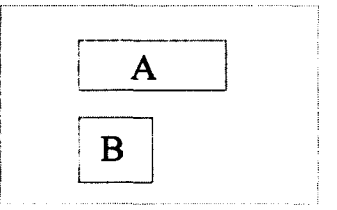
(Image-2) (A, B, 5, 1)

(Image-3) (A, C, 7, 5)

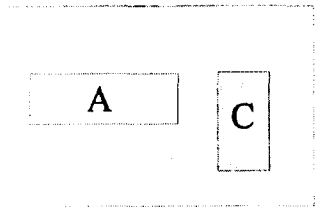
(Image-4) (B, C, 8, 6)



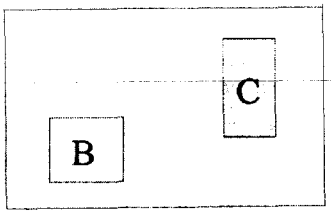
(a) Image-1



(b) Image-2



(c) Image-3



(d) Image-4

(Fig. 7) Four iconic images as example

To create the signature of the four iconic images, we assume that the object signature has 8 bits in length, the approximate-match signature has 16 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}. Table 1, Table 2, and Table 3 list the object, the approximate-match, and the exact-match signatures, respectively. Based on them, we can generate the signatures of the four images as shown in Table 4.

〈Table 1〉 Object signatures

Object	Object signature
A	00010001
B	00100010
C	01000100

〈Table 2〉 Approximate-match signatures

ASR	Approximate-match signature
(A, B, 5, 0)	0000000000000001
(A, B, 5, 1)	0000000000000010
(A, B, 5, 2)	0000000000000100
(A, B, 6, 0)	0000000000001000
(A, B, 6, 1)	0000000000010000
(A, B, 6, 2)	0000000000100000
(A, C, 6, 5)	0000000001000000
(A, C, 6, 6)	0000000010000000
(A, C, 7, 5)	0000000100000000
(A, C, 7, 6)	0000001000000000
(A, C, 8, 5)	0000010000000000
(A, C, 8, 6)	0000100000000000
(B, C, 7, 6)	0001000000000000
(B, C, 1, 7)	0010000000000000
(B, C, 8, 6)	0100000000000000
(B, C, 8, 7)	1000000000000000

〈Table 3〉 Exact-match signatures

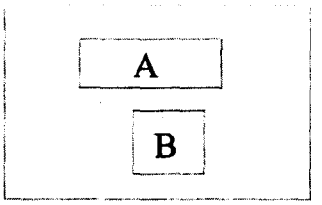
ESR	Exact-match signature
(A, B, 5, 1)	000100000001
(A, C, 7, 5)	001000000010
(B, C, 8, 6)	010000000100

〈Table 4〉 Image signatures for our four example iconic images

Image	IS_{obj}	IS_{app}	IS_{exa}
Image-1	01110111	1111111111111111	011100000111
Image-2	00110011	0000000000111111	000100000001
Image-3	01010101	0000111111000000	001000000010
Image-4	01100110	1111000000000000	010000000000

Fig 6 illustrates a file structure after we insert the four iconic image signatures shown in Table 4. Here we can distribute the four signatures, depending on the suffix value of each object signature. That is, we can store the signature of Image-4 into the group G0 because the suffix of its object signature has '0' bit. We can also store the other signatures into G1 because their suffixes have '1' bit in common. For example, suppose that we have a query to find such an iconic image as Image-Q in Fig 8. To answer this query, we first generate a set of SRS Image-Q in our basic representation as follows:

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



(Fig. 8) Image-Q : A query iconic image

Next, we create the object, the approximate-match and the exact-match signature of Image-Q by using Table 1, Table 2, and Table 3 as follows:

$IS^{Q}_{obj} = 00110011,$
$IS^{Q}_{app} = 000000000011011,$
$IS^{Q}_{exa} = 100000001000.$

Finally, if we require some exact-match answers, we can compare IS^{Q}_{exa} with the three signatures of G1 in the exact-match signature file because the suffix of IS^{Q}_{obj} has '1' bit. This leads to no qualifying signature because no signatures in F_{exa} of G1 satifies the bit pattern of IS^{Q}_{exa}. On the other hand, if we require some approximate-match answers, we can access only the three signatures of G1 in the approximate-match signature file because the suffix of IS^{Q}_{obj} has '1' bit. We obtain two qualifying signatures, i.e., the first and the second in G1, because they contain the bit pattern of IS^{Q}_{app}. Thus, we can access the SRS 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-1 and Image-2, because the SRS of Image-Q is included into both the SRS of Image-1 and the SRS of Image-2.

5. Performance Evaluation

We assume that a pixel-level original image should be automatically or manually transformed into its iconic image including only 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, having its icon name, its size, and its position. For our experiment, we generate the following iconic databases[10].

- 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 5,000.
- A query iconic image contains two to three icon objects.
- The number or queries for the experiment is 200.

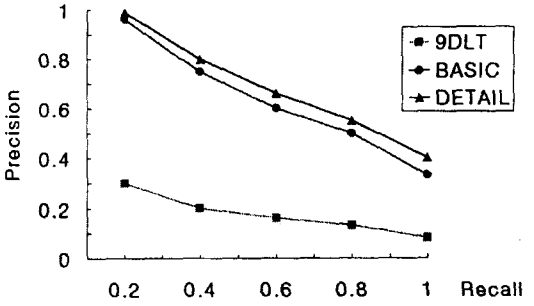
In order to evaluate retrieval effectiveness[11], we make use of recall and precision measures. Let IRT be the number of iconic images retrieved by a given query, IRL be the number of iconic images relevant to the query, and IRR be the number of relevant iconic images retrieved. The relevant images can be determined by computing the similarity between two iconic images, based on both their spatial relationship and their icon size. The recall and precision measures are computed as the following:

$$Re\ call = \frac{IRR}{IRL} * 100$$

$$Pr\ ecision = \frac{IRR}{IRT} * 100$$

When a variety of queries are executed two hundred times, Table 5 shows the average values of IRT, IRL, and IRR in three spatial match representation schemes. In addition, Table 6 shows the retrieval effectiveness of the representation schemes, in terms of precision and recall measures. Our representation schemes improve the retrieval effectiveness considerably, compared to the 9-DLT scheme. Our representation schemes improve retrieval precision by 0.43-0.

56 in the exact match and by 0.2-0.3 in the approximate match, while their recall values are kept higher than those of the 9-DLT scheme.



(Fig. 9) Recall versus Precision

It is also shown that our detailed representation scheme holds about 0.1 higher precision value than our basic one, while their recall values are kept almost the same. This is because our detailed representation scheme reserves all of the possible spatial relationships for an iconic image. Fig 9 also shows the graph of recall versus precision. It is shown from the result that our reprecsentation schemes are much better than 9-DLT scheme. This is because our representation schemes can describe spatial relationships between icon objects accurately.

In order to evaluate the retrieval efficiency, we make use of the probability of false drops[4, 9]. Let N be the number of iconic images (i.e., the number of signatures), Ms be the number of qualifying signatures, and Mf be the number of false drops. The false drop probability, Fd, can be computed as the following:

$$Fd = \frac{Mf}{N-(MS-Mf)}$$

Table 7 shows the retrieval efficiency of our retrieval method for spatial match representation schemes, in terms of Fd. The retrieval efficiency for our representation schemes is slightly decreased, compared to that for the 9-DLT. Namely. Fd is increased by up to 0.005 in the exact match, and by up go 0.008 in the approximate match. Moreover, Table 7 shows the storage overhead(SO) of our retrieval method, in

〈Table 5〉 IRT, IRL, and IRR of spatial match representation schemes

	Exact			Approximate		
	IRT	IRL	IRR	IRT	IRL	IRR
9DLT	51.71	12.85	6.46	100.76	12.85	8.53
BASIC	15.84	12.85	8.90	38.56	12.85	10.97
DETAIL	11.51	12.85	7.90	28.18	12.85	10.93

〈Table 6〉 Retrieval effectiveness of spatial match representation schemes

Retrieval effectiveness	9DLT Scheme		Our Basic Scheme		Our Detailed Scheme	
	Approx.	Exact	Approx.	Exact	Approx.	Exact
Precision	0.09	0.13	0.29	0.56	0.39	0.69
Recall	0.67	0.50	0.86	0.70	0.85	0.62

〈Table 7〉 Fd and SO of our retrieval method for spatial match representation schems

Retrieval measures	9DLT Scheme		Our Basic Scheme		Our Detailed Scheme	
	Approx.	Exact	Approx.	Exact	Approx.	Exact
Fd	0.032	0.033	0.039	0.038	0.040	0.034
SO(Mbyte)	0.23	0.67	0.24	1.07	0.24	2.76

terms of signature file size. It is shown from the results that our representation schemes require up to four times larger signature storage in the approximate match than the 9-DLT one, while they require almost the same signature storage in the exact match. Our detailed representation especially requires considerably great storage overhead.

6. 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 etrieving relevant multimedia documents. The pupose of our paper is to provide effective representation and efficient retrieval of images after a pixel-level original image is transformed into its iconic image. For this, we proposed our spatial match representation schemes so as to support content-based image retriveval in an effective way. Our representation schemes accurately described spatial relationships between icon objects because they could represent the icon object as a rectangle and make use of precise positional operators. To accelerate searching, we also designed our efficient retrieval method based on a two-dimensional signature file organization.

In order to prove the superiority of our representation schemes 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 schemes 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. We also showed that our detailed scheme outperformed our basic one on the precision, while their recall values were kept almost the same. However, our detailed one required considerably larger signature storage in the approximate match, compared to our basic one. For further work, our representation schemes suould be applied to real application areas using iconic images, proving the efficiency of our schemes in these areas.

References

[1] S. K. Chang and 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's 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 model logic of time intervals," Journal of Association Computing Machinary, 38:935-962, 1991.

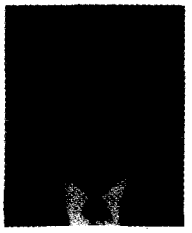
[7] A. D. Bimbo, M. Campanai, and P. Nesi. "A three-dimensional iconic environment for image database qurerying," IEEE Transactions on Software Engineering, 19(10):997-1010, 1993.

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

[9] J. K. Kim and J. W. Chang. "A two-dimensional dynamic signature file methods," In Proc. of the Int'l Symposium on Advanced database Technologies and Their Integration, pages 63-70, Nara, Japan, 1994.

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

[11] G. Salton. "Automatic Text Processing," Addison-Wesley, 1989.



장 재 우

1984년 서울대학교 공과대학 전자계산기공학과 졸업

1986년 한국과학기술원 전산학과 석사학위 취득

1991년 한국과학기술원 전산학과 박사학위 취득

1991년 3월~현재 전북대학교 공과대학 컴퓨터공학과 부교수

1996년 1월~1997년 1월 미국 미네소타 대학교 연구교수

관심분야: 멀티미디어 데이타베이스, 멀티미디어 정보검색

Jaideep Srivastava

1983년 B.Tech. in Computer Science, Indian Institute of Technology, kampur, India

1985년 M.S. in Computer Science, Univ. of California, Berkeley

1988년 Ph.D. in Computer Science, Univ. of California, Berkeley

1994년~현재 Assocate Professor, Dept. of Computer Science, Univ. of Minnesota, Minneapolis, USA

관심분야: 멀티미디어 데이타베이스