INTELLIGENT ICONIC PICTORIAL DATABASE SYSTEM

Xiaoming Zhou (M.Sc)

A thesis submitted in partial fulfillment of the requirements for the degree of

Ph. D in Computer Science

National University of Singapore

2005

ACKNOWLEDGMENT

The author would like to express his most sincere gratitude to his supervisors, Dr. Ang Chuan Heng and Prof. Ling Tok Wang, for the excellent professional guidance and constant assistance rendered in making this thesis possible. The great care and consideration with which they provided outstanding supervision and friendly patience are greatly appreciated and for which the author shall be forever grateful.

The author wants to thank the management of Sybase for the support and understanding of the academic interests. Sybase has not only partially provided the financial assistance for the continued study, but also sponsored many of the academic trips during the last 6 academic years.

The author is also greatly indebted to many staffs in School of Computing, National University of Singapore, for their kind helps in solving many problems during the research. In addition, the author would like to express his appreciation for the comments and hints provided by the anonymous reviewers. They certainly improve the quality of the thesis.

Last, and certainly not least, the author wishes to give very special thanks to his patient and understanding wife and parents for their ever-present encouragement and for the moral and practical support given over the years. Their support has greatly contributed not only to the quality of the research but also in making the effort enjoyable and rewarding. Indeed, their loving support truly made it all possible.

Many friends, too numerous to mention, have given so freely of their time, encouragement and support.

The gratitude to all is beyond measure.

Table of Contents

Summary

Pictorial databases become more and more popular recently. However, in pictorial databases, the real images always require large storage and are not easy to be retrieved. Therefore, "content-based" databases retrieval attracts more and more attentions in these years. Instead of using the original full-sized images, content-based retrieval works by matching the query against a representation of the content of the image. This representation of the image is a symbolic version of the objects in the picture and features extracted from the original picture. Normally, the features of the corresponding pictures include two forms: visual features (such as color, texture, and shape) and relation features (spatial relations among the objects in a picture). In this thesis, we analyze two-dimension (2D) spatial relation representation, similarity measure, indexing, and retrieval process.

We introduce some different spatial relation representations which include using two one dimension (1D) spatial relations to represent 2D spatial relations, using one heuristic 2D spatial relation to represent and index 2D pictures, and using systematic topological and orientation relations to represent 2D spatial relations. We propose an integrated spatial relation representation, which is called Intrinsic Orientation and Topological (IO&T) relation representation, to combine topological relation and orientation relation. We also introduce an Augmented Orientation Spatial Relation (AOSR) representation to cover the missing information in IO&T approach.

Based on the proposed representations, we discuss some similarity measures to index and compare pictures in iconic pictorial databases. Some automatic picture matching and similarity retrieval approaches are addressed individually. Since

similarity is a fuzzy concept for different users and applications, to improve the usability of a retrieval system, we involve users in the interactive retrieval process. Some indexing approaches and retrieval procedures are discussed to capture the subjective information hidden in individual user's feedback to fill the gap between the objective system and individual retrieval process. In the end of the thesis, we explore some interesting topics such as multipoint feedback and dynamic weighting in the interactive similarity retrieval process.

Our research covers most of the areas of an intelligent pictorial database system that includes iconic image representation, similarity measure, index for the similarity retrieval, and automatic and interactive retrieval process. Some prototyping experiments are used to demonstrate the feasibility of our proposed approaches. We also highlight some interesting works to be done in the future.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 The description of pictures

Pictorial similarity retrieval has generated a great deal of interest due to the increase in the popularity of multi-media applications. In a traditional database system, pictures or images are handled as binary large objects (blobs). For the purpose of retrieval, each blob is associated with a manually entered textual description (annotated blob). The retrieval of blob is conducted via its textual description using text retrieval methods. The blob (image/binary) data type is typically not an index-able data type in most commercial databases such as Sybase, MySQL, and Microsoft SQL etc.

However, recently, there is considerable research to develop content-based pictorial retrieval, in which blob will become an index-able data type (for example, an early framework implemented in a commercial database like Oracle8i [Anna00], and a QBIC [Flic95] solution proposed by IBM [http://wwwqbic.almaden.ibm.com/]. Although the experimental feature in some of commercial databases (such as Oracle and DB2) is not really matured, there are more prototyping applications developed in academia recently. Typically, these applications characterize pictures as feature vectors in high-dimensional spaces. A user submits a query picture to a search engine, and the search engine returns pictures that are similar to the query picture. Many existing pictorial retrieval systems (such as [Anna00], [Bach96], [Flic95], [Huan03], [Peuq87], [Smit96] etc) use low-level features for pictures retrieval. The low level features are mostly visual features such as color, texture, and shape etc. These features are useful in many applications that do not require much intelligence. However, the recent progress of image understanding in various domains prompts for the transition to the next level of pictorial retrieval, which is the retrieval of pictures by objects identified. At this level, a pictorial retrieval system is to support higher-level queries involving objects. One of the most important methods for discriminating objects in a pictorial database is the perception of the objects and the spatial relations that exist among them [Chan00], [Disc01], [Gudi98], [Lee92], [Nabi96], [Smyt03]. The spatial data of pictures should be stored in appropriate representations so that users can easily retrieve, visualize, and manipulate pictures. The effectiveness of this representation depends on the correctness and the usefulness of the spatial relation feature chosen. In the following discussion, we concentrate on spatial relation features.

There are basically two ways to describe a spatial relation: the quantitative description and the qualitative description.

In the quantitative description, the magnitudes must be provided. Exact numbers (such as coordinates) are always used to describe the location of objects. By comparing the numbers associated with the objects, their spatial relations can be figured out quantitatively. The advantages of quantitative representation are objectiveness and accuracy.

Another trend in pictorial representation and retrieval is toward the qualitative approach. A qualitative description focuses on the spatial relation of an object with its surrounding objects, and the retrieval is done by matching the related descriptions. Although qualitative information is often mistaken to be vague or inexact, it can be more efficient and provide more meaning than pure quantitative information sometime. For example, saying that Singapore lies at latitude x degrees north and longitude y degree east is a piece of exact quantitative information about the location of Singapore, but very likely it does not resonate with the spatial knowledge of the average reader. On the other hand, saying that Singapore is at the south of Malaysia is cognitively more immediate. Therefore, qualitative answers often contain richer meaning than quantitative data since the description is context-dependent. Many researchers pay their attention to the qualitative approach in these years, and they have proposed many different approaches in areas such as GIS, CAD and multimedia etc. The qualitative representation and retrieval approach is also related to many scientific fields such as pattern recognition, database, HCI, and information retrieval etc.

In general, both qualitative and quantitative representations have their applications. However, the objective of representation is for the ease of retrieval. For pictorial database retrieval, approximate similarity retrieval is more applicable comparing to exact matching. Therefore, qualitative representation has its advantage. In addition, suitable quantitative representation can also play an important role to make up the subjective-ness nature of a qualitative representation. Our research will not rule out the possibility of integrating both qualitative and quantitative representations.

Before the discussion of our research in this thesis, we will have a bird's eye view of what is needed for a pictorial database system.

1.2 Architecture of a pictorial database system

Pictorial information retrieval is different from the text information retrieval in that the latter is based on exact text matching while the former is based on a fuzzy matching between a pictorial query and the pictures in the pictorial database. A good picture representation not only saves local storage, but also improves the efficiency of picture retrieval. Figure 1.1 shows an architecture diagram for a pictorial database system.

Conceptually, the database is partitioned into three parts: a real image repository (where the original full-sized images are stored), an iconic image repository (where an iconic version ("thumbnail") of each image is kept) and the symbolic images repository (which holds the spatial relations of the icons appearing in each picture). The real full-size pictures are always stored in a backup storage whereas the symbolic images are stored locally. The iconic images can also be stored locally when they are presented to the users for the selection of query pictures.

The picture retrieval process is as follows: First, the user submits a pictorial query to the system. The symbolic image transformer converts the pictorial query into the corresponding representation that is at the same abstraction level as the pictures in the symbolic image repository. Then a retrieval engine will be used to retrieve similar pictures from the symbolic image repository. The retrieval results can be symbolic pictures or real full-sized pictures according to users' requirement. If the retrieval engine is replaced by an insertion-engine, the user may also add more pictures into the database.

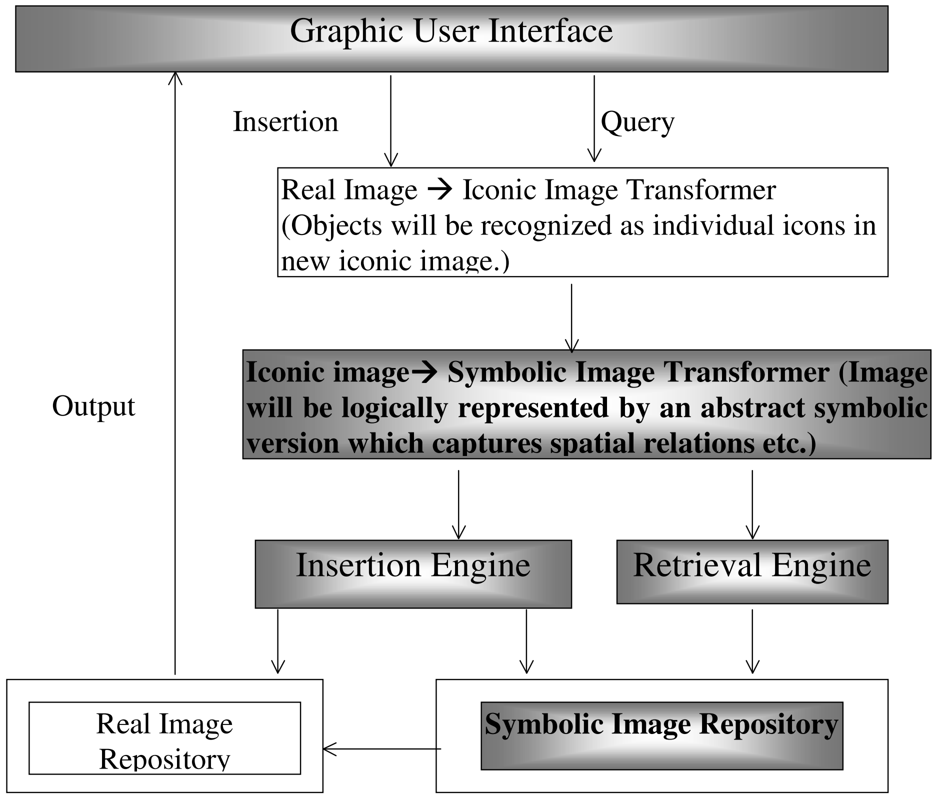


Figure 1.1. Architecture of a pictorial database system

1.3 Our research

Our research focuses mainly on "Iconic Image to Symbolic Image Transformer", "Insertion Engine", and "Retrieval Engine". We also explore some indexing approaches for "Symbolic Image Repository", and the interactive retrieval process for the "Graphic User Interface". The "Real Image to Iconic Image Transformer" is more on Pattern Recognition area [Bhan94] and it is beyond the scope of our topic. The "Real Image Repository" is not our focus.

The main objective of our research is to study some existing spatial relation representations and retrieval processes in different areas and explore a general intelligent pictorial retrieval process without focusing on any specific application area. As an abstract model for spatial relations representation and retrieval process is used, such retrieval process can be applied to different areas by adding specific requirements of different applications.

Since spatial relation representation for approximate similarity retrieval is normally a fuzzy concept, in order to build a good representation model, we need to consider some important design issues: *Completeness* requires the model to include enough objective information instead of partial or subjective information. *Conciseness* helps to save storage space and improve efficiency by avoiding redundancy. *Consistenc*y denotes the invariance under the translation and rotation of pictures and demonstrates the robustness of a representation system directly.

The efficiency of a similarity retrieval algorithm depends on a good representation model as well as the notion of similarity. As the requirements for similarity retrievals vary from applications to applications, it is quite difficult to define a good metrics system for similarity retrieval in general. A good metrics system must be *effective* and *objective* so that users can retrieve the wanted answer smoothly and accurately without being misled into a wrong conclusion. Finding an effective similarity metrics is the most challenging part in the design of the similarity retrieval algorithm.

Besides a representation model and a metrics system based on the representation model, a good indexing structure is also very important in the organization of a huge pictorial database. The indexes must be easy to build and maintain. At the same time, the indexing structure should cater for the efficient retrieval for different objectives. In particular, an easy to follow retrieval process can be built upon the indexing structure to guide users for intuitive and convenient similarity retrieval.

In short, our study covers different areas of a complete retrieval system that includes picture representation, similarity metrics, indexing structure and the retrieval process. In the following, we will give an overview of this thesis.

1.4 Overview

Chapter 2 describes the early development of spatial relation representation from 1D space to 2D space. A representative method 2D string and its picture-matching scheme will be introduced. As a comparison, another representative approach, SHV, will be discussed based on hashing concept. We will also survey some other researches in spatial relation representation and its retrieval measure.

Chapter 3 continues to discuss spatial relations representation in 2D space systematically and introduces a new representation approach, IO&T, which combines intrinsic orientation and topological relations in the picture representation.

Chapter 4 proposes and compares different similarity measures based on the previous discussed representations, and analyzes the cons and pros of these approaches. An object-based retrieval concept is proposed to trim the answer set of similar pictures. This approach makes the similarity retrieval more application oriented and get the users involved in the fully automatic retrieval process.

Chapter 5 introduces an augmented orientation representation approach that provides some information that is missing from the previous representation approaches. It compares the automatic similarity retrieval represented by 2D- PIR, IO&T, and AOSR approaches.

Chapter 6 discusses the interactive similarity retrieval process. A flexible indexing scheme is proposed to support a user-friendly retrieval procedure. This is the current trend of moving to feedback-based system from fully automatic retrieval.

Chapter 7 explores the use of multipoint feedback and dynamic weighting in the interactive similarity retrieval process. Some other interactive similarity retrieval approaches in different area will be used as comparison at the end of this chapter.

Chapter 8 summarizes the major contributions and limitations of the proposed solutions, and discusses some issues to be studied in the future work.

Part of this thesis has been published in [Ang98], [Zhou97], [Zhou00], [Zhou01], [Zhou02], [Zhou04], and [Zhou06] (excluding submitted papers).

Chapter 2

Legacy spatial relation representation and matching

2.1 From 1D to 2D

Early work concerning spatial relation representation starts from 1D space. Allen [Alle83] introduced an interval-based temporal logic, in which, knowledge about time is maintained qualitatively by storing comparative spatial relations between intervals. The thirteen possible spatial relations between one-dimensional intervals were described in his landmark paper on temporal intervals. Figure 2.1 shows the geometric interpretation of these interval spatial relations for 1D objects A and B.

The elegance and simplicity of that approach has inspired several efforts to extend it to 2D spatial dimensions. One way is to make the same kind of distinctions as in the 1D temporal case for two axes of a Cartesian coordinate system. Object boundaries are projected onto the two axes and a pair [x-Relation, y-Relation] is used to give the relative position of objects. Many approaches (such as [Chan87], [Gues89], and [Lee91] etc) have been proposed based on this idea. Among them, 2D string is a typical work that opens a new area for object spatial relation representation. Subsequently, there are many variants and extensions of 2D string developed by other researchers. In the next section, we will take a look at 2D *-string for 2D spatial relation representation and indexing (where 2D *-string refers to variant 2D strings and its typical extensions such as 2D string, 2D C-string, and 2D C+-string etc).

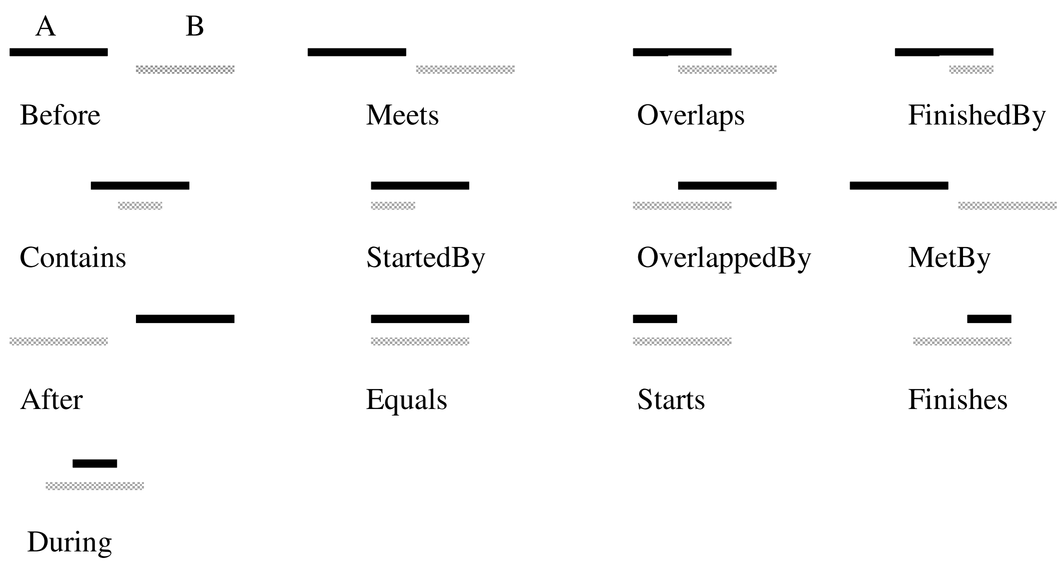


Figure 2.1. Allen's interval spatial relations

2.2 2D *-string approach

Tanimoto [Tani76] suggested the use of picture icons as picture indexes, thus introducing the concept of iconic indexing. Subsequently, Chang et al [Chan87] developed the concept of iconic indexing by introducing the 2D string representation of symbolic pictures. Although their very first approach is quite straightforward, it inspires many further researches along 2D string's direction.

First, for any spatial representation, to represent symbolically a picture, the objects in the original picture must be recognized. In 2D string, these recognized objects are enclosed by minimum bounding rectangles (MBR). For each object, the orthogonal spatial relation of objects with respect to the other objects are generated by their respective MBRs' projections on x-axis and y-axis. Three operators "<", "=", and ":" are employed to denote the "left-right or below-above" (for example, a and b in Figure 2.2), the "at the same spatial location as" (for example, a and d on x-axis in Figure 2.2), and the "in the same set as" (for example, d and e in Figure 2.2) spatial relation in 1D axis. The detail definition of 2D string is as follows:

- Let V be a set of symbols that represent pictorial objects.
- Let A be the set {"=", "<", ":"}.
- A 1D string over V is any string $x_1x_2...x_n$, n>=0, where the x_i's are in V.
- A 2D string over V, written as (u, v) is defined to be

  $(x_1y_1x_2y_2...y_{n-1}x_n, x_{p(1)}z_1x_{p(2)}z_2...z_{n-1}x_{p(n)})$

  where

  $x_1 ... x_n$ is a 1D string over V;

  P: {1, ..., n}--> {1, ... , n} is a permutation over {1, ..., n};

  $y_1... y_{n-1}$ is a 1D string over A;

  $z_1 ... z_{n-1}$ is a 1D string over A.

For example, if we use 2D string to represent the picture in Figure 2.2 given V as {a, b, c, d, e}, the 2D string representing the picture is:

$(a=d:e<b<c, a<b=c<d:e) = (x_1y_1x_2y_2x_3y_3x_4y_4x_5, x_1z_1x_4z_2x_5z_3x_2z_4x_3)$

where $x_1x_2x_3x_4x_5$ is a d e b c;

$x_1x_4x_5x_2x_3$ is a b c d e;

p is 1 4 5 2 3;

$y_1y_2y_3y_4$ is = : < <;

$z_1z_2z_3z_4$ is < = < :.

de		
	b	c
a		

Figure 2.2. A symbolic picture

At this stage, we can see that the symbolic picture that preserves the spatial relations among MBRs of the original picture is encoded as a 2D string. Therefore, the problem of pictorial information retrieval becomes the problem of 2D subsequence matching. This approach allows a natural way to construct iconic indices for pictures (Note: In 2D string, ":" can be omitted and will be omitted according to [Chan87] because it is actually the same as "=" after the projection to 1D axis i.e. x axis or y axis. Therefore, only two spatial operators "<" and "=" are used in 2D string in fact).

Unfortunately, not all spatial relations between real objects can be represented by their MBRs. For complex pictures with many non-rectangle objects, Chang et al proposed a method called orthogonal spatial relations to segment the objects into smaller objects to resolve this issue.

However, Lee and Hsu [Lee90] pointed out that two spatial operators of 2D string (i.e. "<" and "=") were not sufficient to give a complete description for pictures of arbitrary complexity. By adding some new operators, they categorized the spatial relations between two enclosing rectangles into 13 types as shown in Figure 2.3. This classification is quite similar to that of Allen. They also proposed a new cutting mechanism for the extended string - 2D C-string to reduce significantly the number of subparts for objects and the lengths of the strings for representing the pictures. A 2D C-string is defined as a 5-tuple (S, C, R_g, R_l, “( )”), where

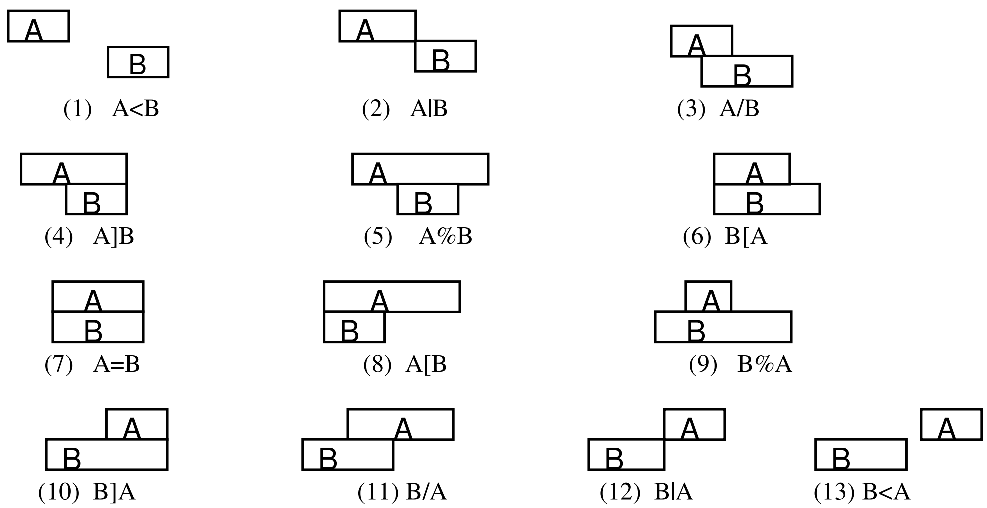


Figure 2.3. The 13 types of spatial relations in one dimension (x axis)

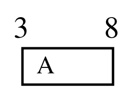
- S is the set of symbols in symbolic pictures of interest;
- C is the cutting mechanism, which consists of cutting lines at the points in the overlapping of x- and y-projections, respectively;
- R_g = {“<”, “|”} is the set of global relational operators;
- R_l = {“=”, “[”, “]”, “%”} is the set of local relational operators;
- “( )” is a pair of separators that is used to describe a set of symbols as one local body.

2D C-string representation captures more spatial relations among the objects in a symbolic picture. However, it still ignores information about relative sizes and locations of objects. Huang and Jean [Huan94] argued that distortion could occur when information of sizes and locations of objects is missing. An example in Figure 2.4 was used to show how distortion could occur when symbolic pictures were reconstructed from 2D C-strings. In their example, both pictures f1 and f2 are projected along the x-axis and represented by the same 2D C-string, A<B%C, which is reconstructed into the picture f3. It is obvious that there are significant differences among symbolic pictures f1, f2, and f3 in terms of relative locations and size of objects A, B, C. Although this example only shows the distortion in the x-axis, distortion in the y-axis can also occur as in the x-axis. According to Huang and Jean's analysis, the problem of distortion was due to ignoring the metric information for the symbolic picture. Without relative metric information, many important queries can not be answered. Therefore, they proposed a 2D C+-string representation scheme by including relative metric information about the picture into the 2D C-String to resolve this problem.

Basically, the 2D C+-string associates 2D C-string with some metric information as follows:

1. Object A with size s, denoted as A_s, where s= end(A) -begin(A),

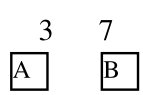
For example,



is represented by A_5.

2. Operator "<" with distance d between object A and B, denoted as $A <_d B$.

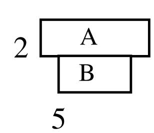
For example,



is represented by $A<_4B$.

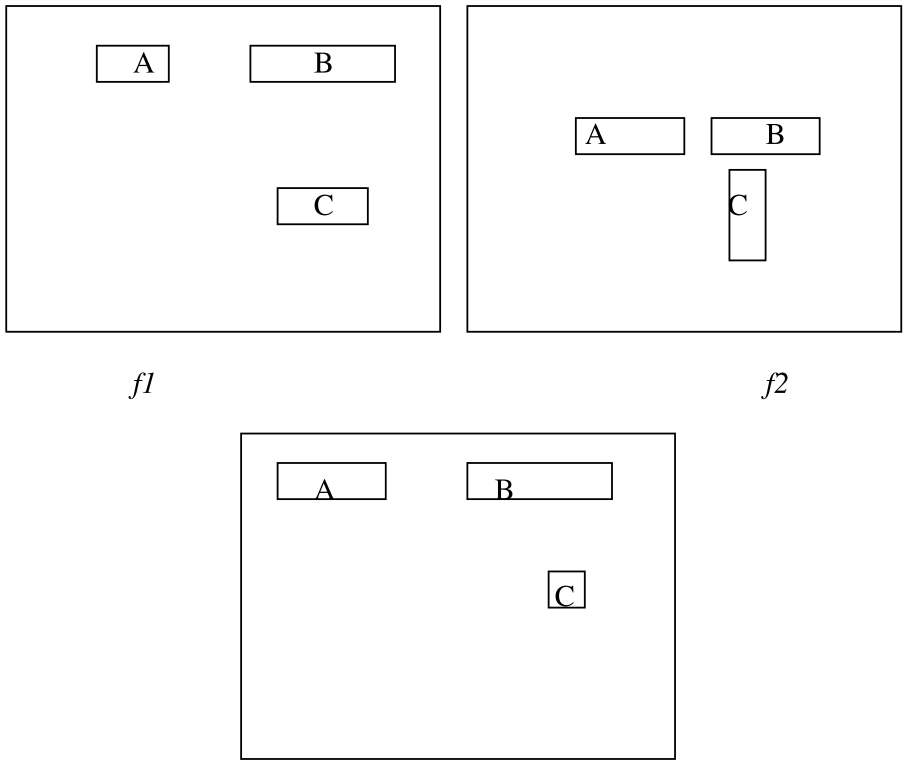
3. Operator "%" with distance d between object A and B, denoted as $A \%_d B$.

For example,



is represented by $A\%_3B$.

4. Other operators: no metric information is needed.



f1 *f2*

f3

Figure 2.4. Symbolic pictures f1, f2, and f3 represented by the same 2D string

Huang and Jean claimed that 2D C+-string could provide the users with greater expressive power to describe pictures more accurately in browsing and visualization. It is obvious that the ambiguity will be reduced with more quantitative information included. However, this kind of quantitative information will cause other problems (such as inflexibility), which is not invariant under transformation and rotation, in particular, the size of object is quite subjective. Therefore, we will not explore further in this direction.

2.3 Picture matching by 2D string matching

As we mentioned earlier, 2D string is one of the many representations that use two one dimensional spatial relations to represent 2D space spatial relations. In this section, we will see how pictures can be retrieved using 2D string matching.

There are many researches involving 2D *-string retrieval (such as [Chan91a], [Chan91b], [Lee91], [Chan92], [Wu94], and [Zhan03] etc). Most of them are based on 2D String ranking. In the following, we will introduce the original 2D string's matching approach for picture matching.

2D string representation makes it simple for sub-picture matching with ranking. The rank of each symbol in a string u is defined to be one plus the number of "<" preceding this symbol in u. Therefore, if we denote the rank of symbol s by r(s), the strings "ad<b<c" and "a<c" have ranks of various symbols as shown in Table 2.1.

Table 2.1. Ranks of Strings

string v						string u		
a	d	<	b	<	c	a	<	c
1	1		2		3	1		2

According to this ranking, some 2D string matching notations can be explained as follows:

- A sub-string where all symbols have the same rank is called a local sub-string.
- A string u is contained in a string v, if u is a subsequence of a permutation string of v.

- A string u is a type-i 1D subsequence of string v, if u is contained in v, and if a1 w1 b1 is a sub-string of u, with a1 matches a2 in v and b1 matches b2 in v, then

  (type-0) r(b2) -r(a2) >= r(b1) -r(a1) or r(b1)-r(a1)=0

  (type-1) r(b2) -r(a2) >= r(b1) -r(a1) >0 or r(b2)-r(a2)=r(b1)-r(a1)=0

  (type-2) r(b2)-r(a2)=r(b1)-r(a1)

We can extend the definition to 2D subsequence as follows. Let (u, v) and (u', v') be the 2D string representation of f and f', respectively. (u', v') is a type-i 2D subsequence of (u, v) if u' is type-i 1D sequence of u, and v' is type-i 1d subsequence of v. We say f' is a type-i sub-picture of f.

In Figure 2.5, f1, f2, f3 are all type-0 sub-pictures of f; f1 and f2 are type-1 sub-pictures of f; only f1 is a type-2 sub-picture of f. The 2D string representations are:

f (ad<b<c, a< bc< d)

f1 (a< b, a< b)

f2 (a<c, a<c)

f3 (ab<c, a<bc)

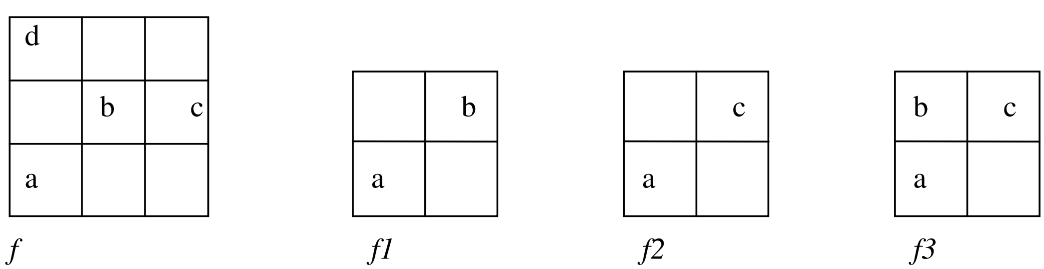


Figure 2.5. Picture matching example

To determine whether a picture f' is a type-i sub-picture of f, we need only to find out whether (u', v') is type-i 2D subsequence of (u, v). The problem of picture matching thus becomes a problem of 2D string matching.

2.4 Integrated 2D spatial relation

2D string method can only handle pictures with rectangular objects. Although Chang et al [Chan87] proposed to segment the pictorial objects orthogonally to simplify the relations, some symbolic pictures are still too complex to be represented as 2D strings and they will cause some problems in spatial reasoning. These problems were alleviated when Lee and Hsu [Lee90] proposed 2D C-string that includes a new cutting mechanism with some new operators. However, some legacy problems as follows are still not overcome.

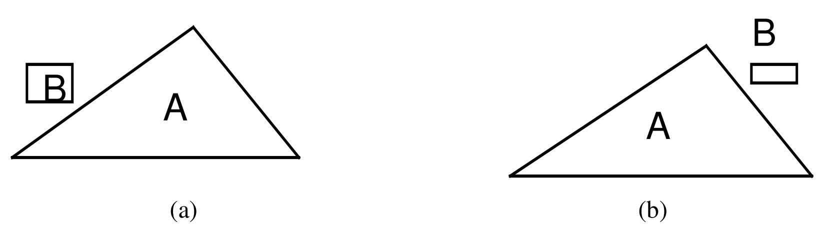


Figure 2.6. Symbolic pictures represented by the same 2D C-string

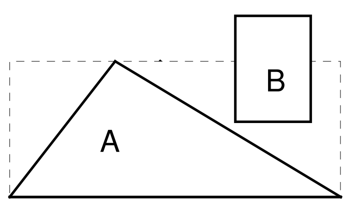


Figure 2.7. The MBRs intersect each other, but the objects do not

In the pictures described by 2D C-Strings, the minimum bounding rectangles (MBRs) of object A and B are used to define the spatial relation between objects.

However, we can hardly describe satisfactorily the spatial relation of the objects in terms of the spatial relation between their corresponding MBRs. For example, in Figure 2.6(a) and 2.6(b), the projection of B along the x-axis and the y-axis are proper subsets of those of A. They are the number 5 spatial relation A%B in Figure 2.3 [Lee90]. Their corresponding 2D C-strings are both (A%B, A%B). However, the relative positions of A and B are entirely different. Even when the sizes of the objects and the distances between them are included, this ambiguity problem still cannot be resolved completely.

In Figure 2.7, even though the objects A and B are disjoint, their MBRs intersect. Therefore, the topological relation between the objects can be quite different from the spatial relation of their respective MBRs, which means that simply using two 1D spatial relations to represent 2D space spatial relation may lose some information. As such, in [Zhou97], we introduce a different categorization of 2D spatial relations that gives a more succinct and precise definition with no information loss.

In the new integrated categorization of spatial relations in 2D space, we still adopt Allen's 1D interval concept. However, instead of using MBRs, we use the centroids and boundaries of objects to describe the 2D spatial relation between objects more accurately.

In the following discussion, we will use a triplet (O[i],O[j],r[i,j]) to represent a pair-wise spatial relation r[i,j] between two objects O[i] and O[j]. We ignore quantitative information, such as the sizes of the objects and the distance between two centroids. There are 41 spatial relations in 2D space in the integrated categorization as shown in Figure 2.8, and they are numbered from 0 to 40. This categorization is an extension of the concept of the 9DLT (9-Directional Lower Triangular) used by Chang and Lee [Chan91a]. This categorization is to overcome the ambiguity of spatial relations described by MBRs used to mimic spatial relations in 2D space. We consider 8 directions combined with 5 types of topological relations and a special category (the same position spatial relation). The categorization rules are described in Algorithm 2.1.

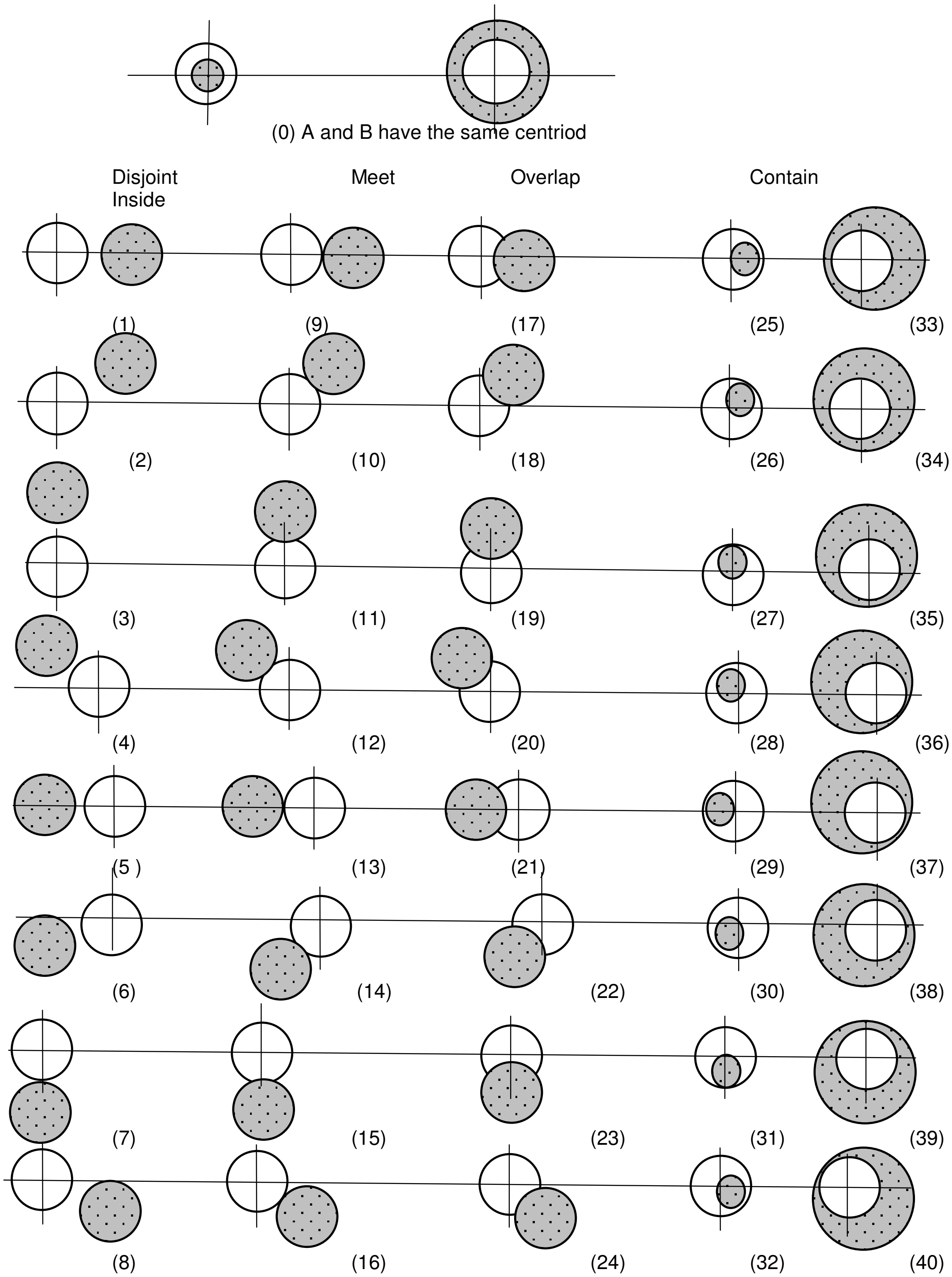


Figure 2.8. The 41 types of spatial relations in 2D space

Algorithm 2.1: Spatial Relation Categorization

Input: Two symbolic objects O[i] and O[j] with boundaries and centroids

Output: The spatial relation r[i,j] between O[i], O[j]

begin

1. *Calculate the distance DIS[i,j] between the centroids;*
2. *If DIS[i,j]=0 then return 0;*
3. *Calculate the direction r[i,j] from the centroid of O[i] to the centroid of O[j];*

   / There are eight possible directions in which the centroid of O[j]can be located with respect to that of O[i]: E, NE quadrant, N, NW quadrant, W, SW quadrant, S, and SE quadrant with the corresponding values ranging from 1 to 8 respectively*/*

   / Check the boundaries:*/*
4. *If O[i] and O[j] are disjoint, return r[i,j];*
5. *If O[i] and O[j] are tangent, return r[i,j]+8;*
6. *If O[i] and O[j] are partially-overlapping, return r[i,j]+16;*
7. *If O[i] totally contains O[j], return r[i,j]+ 24;*
8. *If O[j] totally contains O[i], return r[i,j] + 32.*

end

This classification is more intuitive as compared to the 2D string approach. It also allows us to match pictures easily as described in the following sections.

2.5 Represent and retrieve spatial relations by hashing

Chang et al proposed 2D string to convert the problem of pictorial information representation and retrieval to a problem of 2D string construction and string matching. With the integrated categorization, a symbolic picture can be represented by a set of hash values. This method combines the picture representation with the picture retrieval, and avoids the ambiguity problems that exist in other methods. In the following, we describe how to use a Set of Hash Values (SHV) for picture representation.

2.5.1 SHV representation

Let us consider a large pictorial database with objects taken from a set G. Assume there are N objects in G and a set S of n objects in a picture P, with n<<N. P can be represented by n(n-1)/2 triplets of (O[i],O[j],r[i,j]), where O[i], O[j]$\in$S and r[i,j]$\in$[0,40]. We want to map P to a set of hash values, SHV in short, and use it to represent P uniquely.

Cichelli [Cich80] was the first one to propose an algorithm to map a set of n distinct triplets (O[i], O[j], r[i,j]) to a set of n consecutive addresses, where O[i], O[j] are two symbols and r[i,j] is a positive integer. His algorithm performs an exhaustive search to find the appropriate values for the symbols. Cook and Oldehoeft [Cook82] improved upon Cichelli's method when they proposed Algorithm 2.2 to handle more than one triplet in the search for suitable values for the symbols. Algorithm 2.2 assigns values for symbols such that the mapping h allows only a small number of blank slots in the address space.

Algorithm 2.2: Hash Value Generation (Cook and Oldehoeft [Cook82])

Input: A set R of (O[i],O[j],r[i,j])

Output: A set of values for symbols such that the hash function h() is one to one:

h(O[i],O[j],r[i,j])=r[i,j]+ associated value of O[i]+ associated value of O[j]

begin

1. *Compute the frequencies of the symbols in the triplets. If a symbol appears as both the first and second element of the same triplet, it is assigned a large frequency value.*
2. *Order the symbols by decreasing frequency, resolving ties arbitrarily.*

3. *Replace each triplet (O[i],O[j],r[i,j]) with (O[j],O[i],r[j,i]) if O[j] precedes O[i] in the symbol ordering.*
4. *Use the second symbol in each triplet as the key, sort the list of triplets in descending order.*
5. *Assign symbol values beginning with the first triplet. For each group of triplets with the same second symbol, find a symbol value assignment for the second symbol.*

end

The first symbol is assigned the value 0. Step 5 of Algorithm 2.2 searches for a value for the next symbol that maps the group of triplets to distinct empty slots. If it finds a value, it assigns the value to the next symbol and places the group of triplets in the table. Otherwise, it backs up to the previous group of triplets, un-assigns symbol values, removes the associated mapping table entries and searches again. The complexity of the hash value generation is O(n) for the best case, but it would become much bigger than O(n) in the worst case.

Since all symbols in the representation of a picture appear with the same frequency in all the n(n-1)/2 triplets, their values can be assigned easily with Algorithm 2.3 as follows:

Algorithm 2.3: Symbol Value Mapping

Input: The set G of all object symbols

Output: The set of symbol values SV={V[i]| 1<=i<=N}

begin

1. *Arrange the symbols in any linear order;*
2. *Assign symbol values V[i] to O[i] as follows: V[1]=0, V[2]=1, and V[i]=V[i-1]+V[i-2]+1.*

end

The complexity of Algorithm 2.3 is determined by that of the Step 2 which is O(n). It is more efficient than Algorithm 2.2. In Algorithm 2.2, symbol values have to be re-assigned and hash values of all triplets have to be recalculated when new symbols are added. When Algorithm 2.3 is used, only the associated value of the new symbol needs to be calculated using the values of the last two symbols in G. The symbol values of the original symbols and the hash values of the original triplets will not be affected. The set of symbol values (SV) can then be used to generate easily the set of hashing values (SHV) using Algorithm 2.4.

Algorithm 2.4: Construct the SHV of a picture P

Input: A picture P {(O[1], O[2], r[1,2]), (O[1], O[3], r[1,3]),...} and the set of symbol values SV

Output: The SHV of P

begin

1. *For each (O[i], O[j], r[i,j]), interchange O[i] and O[j] if O[j]<O[i] and replace r[i,j] with r[j,i] accordingly;*
2. *For each (O[i], O[j], r[i,j]), compute h(O[i], O[j]) = V[i] + V[j] and hr(O[i], O[j], r[i,j]) = h(O[i], O[j]) + r[i,j]/100;*
3. *Output SHV={hr(O[i], O[j], r[i,j]) | (O[i], O[j], r[i,j])∈P}.*

end

In Algorithm 2.4, h is an integer-valued function and hr is a real-valued function. The spatial relation is encoded in the fractional part of the hash value of the corresponding triplet. Chang and Lee [Chan91b] applied a similar algorithm for image retrieval only whereas we use these hash values for the image representation as well.

2.5.2 SHV picture matching

For picture matching, the system is required to retrieve from the pictorial database those pictures that are the same to the query picture or include the query picture .

Since each picture is just a sequence of hash values, the problem of picture matching becomes the problem of subset matching between the SHV of the given picture and all the SHVs of all pictures in the database. Instead of searching through all the SHVs, an index structure can be used to identify the matched pictures as quickly as possible.

In detail, a table is used to organize the ids of those pictures containing the pairs (O[j], O[k]). Each pair (O[j], O[k]) is hashed to i = V[j] + V[k], the offset into the table of an entry that points to a tree-like structure with 41 branches corresponding to the 41 spatial relations described in Figure 2.8. The branch for r[j, k] in the tree-like structure is a linked list of the ids of those pictures containing (O[j], O[k], r[j, k]). The structure of the index table is shown in Figure 2.9, and the similarity retrieval procedure that makes use of the table is discussed in Algorithm 2.5.

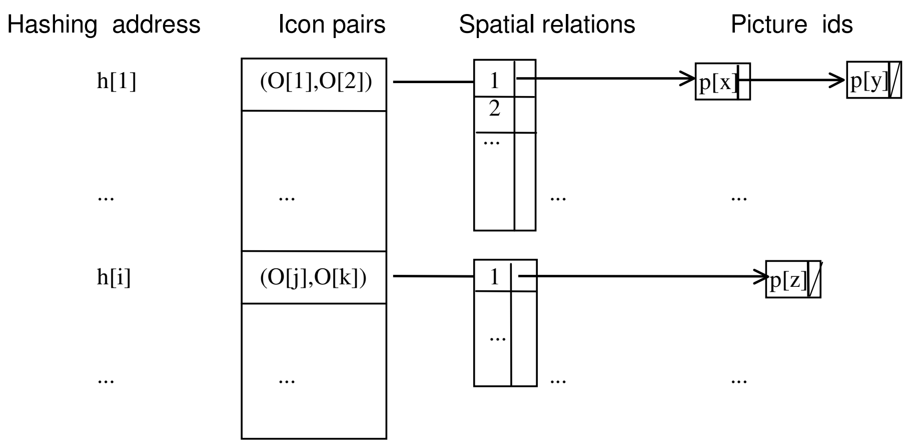


Figure 2.9. Index structure for a pictorial database based on SHV

Algorithm 2.5: Picture matching using SHV

Input: The SHV of size n of a query picture Q

Output: All pictures matched with using SHV

begin

1. *For each hr[i] in SHV, let h[i] = int(hr[i]), and r[i] = (hr[i] - h[i]) * 100;*
2. *Access through the table entry corresponding to h[i] and the r[i] branch to obtain the list L[i];*

3. Calculate L = L[1] ∩ L[2] ∩ ... ∩ L[n].

end

2.5.3 A complete example for SHV

In this section, we give an example to illustrate the generation of SHV and demonstrate how the similarity retrieval can be done based on it.

• Representing a picture by SHV

We use the example picture P in Figure 2.10 to show how its SHV is constructed.

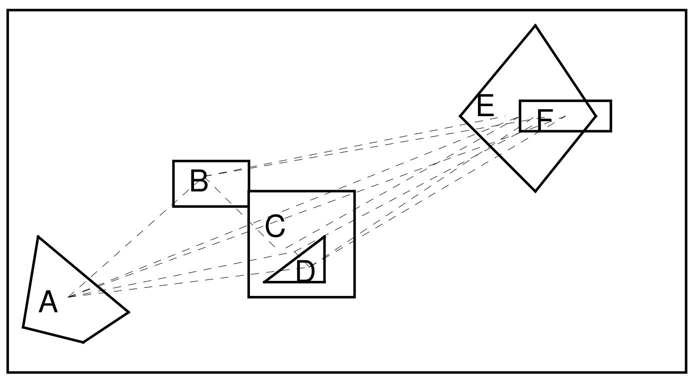


Figure 2.10. An example picture P

The symbol set of picture P is {A, B, C, D, E, F} which is assumed to be G, our universal set of icon symbols. The triplets of P produced by Algorithm 2.1 are {(A, B, 3), (A, C, 3), (B, C, 17), (A, D, 3), (B, D, 9), (C, D, 33), (A, E, 3), (B, E, 3), (C, E, 3), (D, E, 3), (A, F, 3), (B, F, 3), (C, F, 3), (D, F, 3), (E, F, 18)}.

Next, we compute the values for symbols O[i] and icon pairs (O[i], O[j]) by Algorithm 2.3 and the hash values hr(O[i], O[j], r[i, j]) by Algorithm 2.4. There are 6*(6 – 1)/2 = 15 icon pairs to be constructed as shown in Table 2.2.

Table 2.2. Illustrating SHV construction

Symbols O[i] in given order	Symbol value V[i]	Icon pairs (O[j],O[k])	r[j,k]	h(O[j],O[k])=V[j]+V[k]	SHV
A	0	(A,B)	3	1	1.03
B	1	(A,C)	3	2	2.03
C	2	(B,C)	17	3	3.17
D	4	(A,D)	3	4	4.03
E	7	(B,D)	9	5	5.09
F	12	(C,D)	33	6	6.33
		(A,E)	3	7	7.03
		(B,E)	3	8	8.03
		(C,E)	3	9	9.03
		(D,E)	3	11	11.03
		(A,F)	3	12	12.03
		(B,F)	3	13	13.03
		(C,F)	3	14	14.03
		(D,F)	3	16	16.03
		(E,F)	18	19	19.18

• **Example of similarity retrieval**

Assume that the id of the picture in Figure 2.10 is 1 and there are 5 more pictures with ids 2, 3, 4, 5, and 6 in the pictorial database. The index structure of the database is shown in Figure 2.11.

Given a query picture Q which is shown in Figure 2.12, its SHV is {5.09, 8.03, 11.03} of size 3. According to Algorithm 3.5, we have L[1]={1, 4, 6}, L[2]={1, 3, 5}, L[3] = {1}, and L = {1, 4, 6} $\cap$ {1, 3, 5} $\cap$ {1} = {1}.

So only picture 1 that matches the query picture will be retrieved from the database. Although the database can be very large and the corresponding index structure can be very large too, we only pull out 3 lists of picture indices from which the list of similar pictures can be worked out.

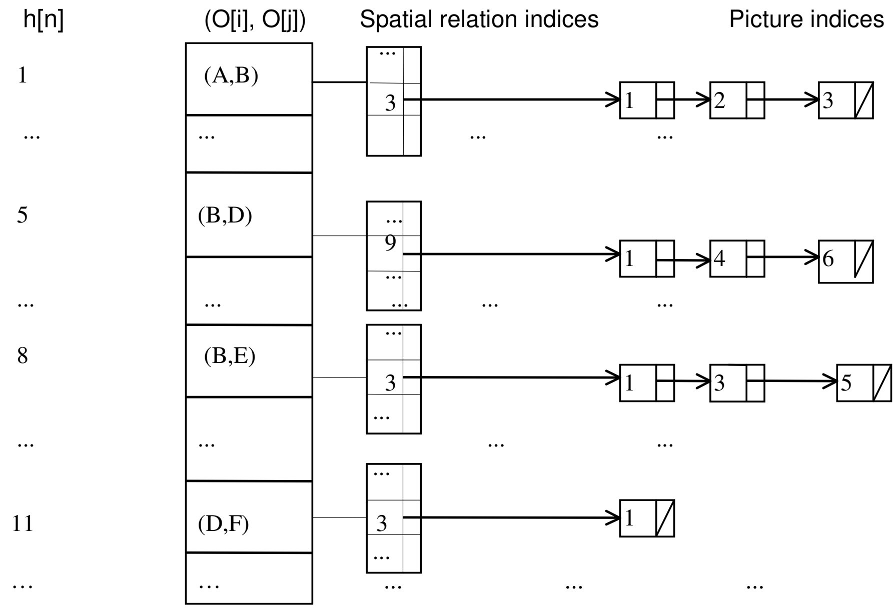


Figure 2.11. Example of the index structure of the database

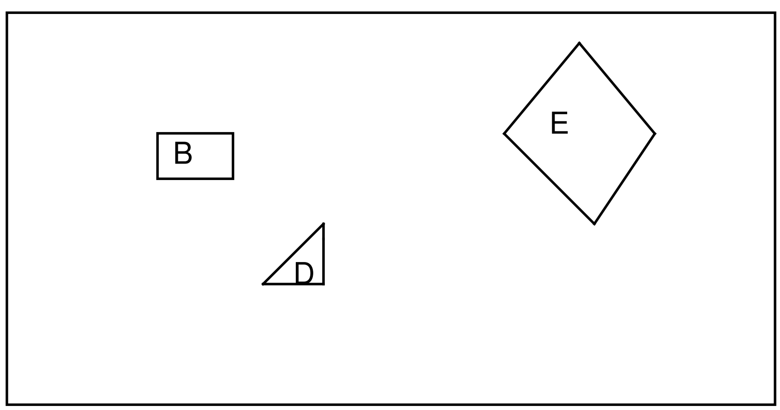


Figure 2.12. Query picture

2.5.4 Comparison of 2D string and SHV representation

In previous sections, we discuss some early researches in spatial relation representation and matching, in particular, 2D string and SHV. Compared with 2D string, SHV can be constructed easily to encode pairs of objects and their spatial relations and it can be used directly to retrieve the relevant pictures through an index structure. It avoids the ambiguity problem in visualization and the need to use complex rules or algorithms in retrieval, the two shortcomings exhibited in other representation methods. However, the categorization of SHV is quite arbitrary and heuristic, in particular, the approach from SHV is more suitable for picture matching instead of similarity retrieval. Although there is an enhancement [Huan04] lately, the new hashing approach in [Huan04] is also not suitable for similarity retrieval. In fact, in most real applications, approximate (similarity) retrieval instead of exact retrieval is more useful. Therefore, we will discuss similarity retrieval in the following chapters.

2.6 Survey and discussion of other spatial relation representations for automatic similarity retrieval

To round up our discussion on legacy picture representations, we look at some other works in this area.

2.6.1 Recursive decomposition approaches

The quadtree introduced by Samet [Same84] is a hierarchical data structure based on the principle of recursive decomposition of space. Basically, a picture will be divided into four equal-sized quadrants, and the quadrants can be subdivided into sub-quadrants, sub-sub-quadrants, and so on. Symbols in the picture will then be attached to the respective quadrants or sub-quadrants, and the spatial relations among symbols can be reflected by their locations in the quadtree.

The R-tree from Guttman [Gutt84] is another hierarchical data structure that is derived from the B-tree. Each node in the tree corresponds to the smallest d-dimensional rectangle that encloses its child nodes. The leaf nodes contain pointers to the actual geometric objects. Its spatial knowledge representation is similar to that of quadtree though the data structure is different.

2D H-string ([Chan88], [Chan96]) is a combination of quadtrees and 2D strings. It can be described in a concise way as follows.

Given an $m \times n$ symbolic picture P: $\{1,2,\ldots,m\} \times \{1,2,\ldots,n\} \rightarrow 2^S$, where S is a set of symbols or corresponding vocabularies, P is subdivided into four quadrants Q1, Q2, Q3, and Q4 which represent NW, SW, NE, and SE, respectively (Figure 2.13. A 2D-H string of a picture P, denoted as 2D-H(P) is defined recursively as

2D-H(P)= $\Downarrow$ P1 P2 P3 P4 $\Uparrow$

Q1	Q3
Q2	Q4

Figure 2.13. Four quadrants of a picture P

where $\Downarrow = \{\downarrow 0, \downarrow 1, \downarrow 2, \downarrow 3\}$ is the set of down-level operators, $\Uparrow = \{0\uparrow, 1\uparrow, 2\uparrow, 3\uparrow\}$ is the set of up-level operators, and Pi=2D-H(Qi) for i=1, 2, 3, 4. The up-level operators determine, in pair with the down-level operators, the local spatial relations between quadrant blocks. Their specific elements in the sets of up-level operators and down-level operators are shown in the pattern of Figure 2.14. For those Qi that can not be represented by one symbol, the subdivision process is repeated. When one of the sixteen types of pattern in Figure 2.14 is reached, and the subdivision stops.

Quadtree is popular in spatial database, but it is hard to use in iconic image database for similarity retrieval.

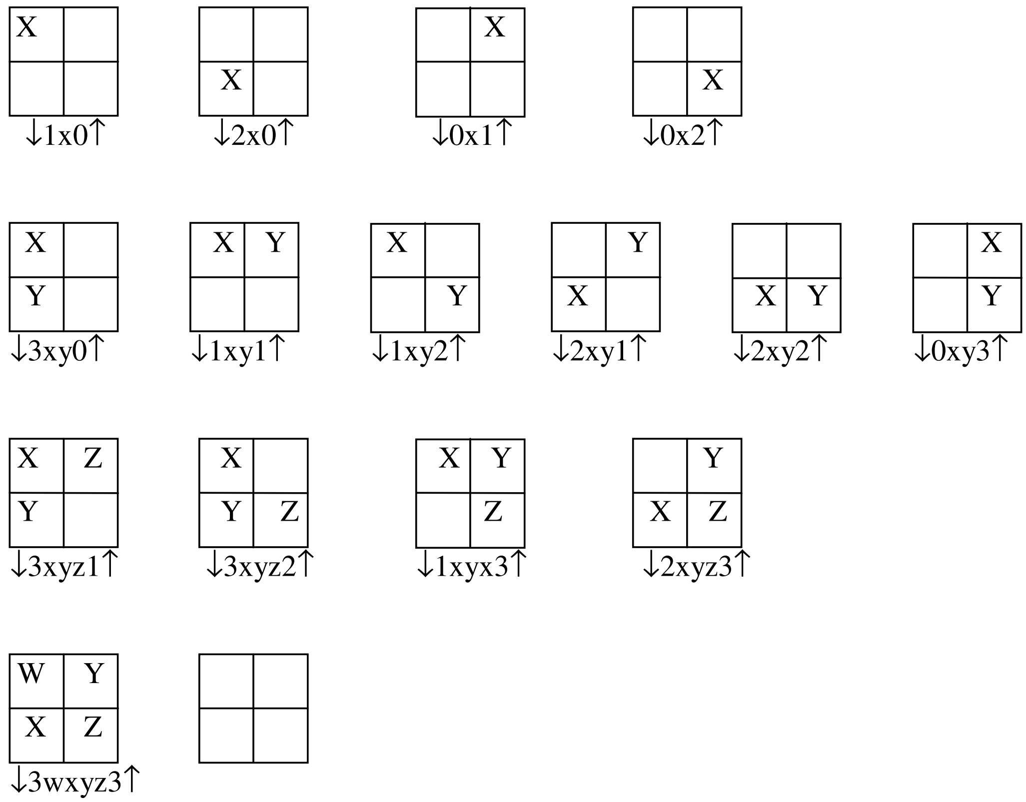


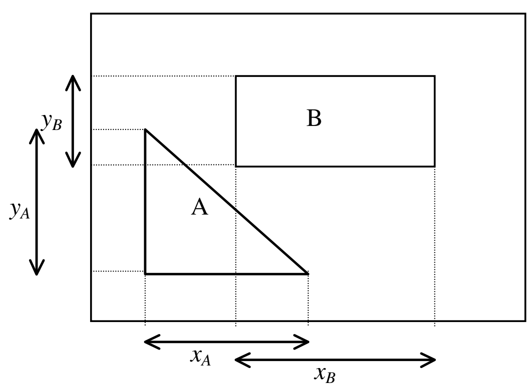
Figure 2.14. Sixteen basic types of patterns

2.6.2 Interval algebras

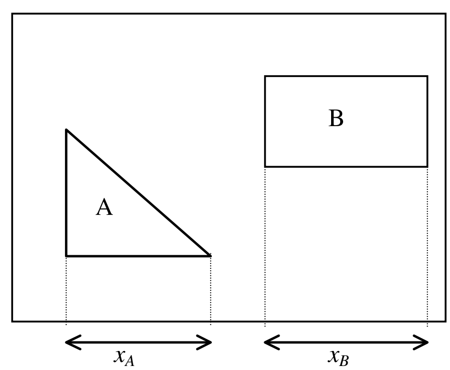
As we have mentioned in previous chapter, Allen [Alle83] introduced an interval-based temporal logic that has inspired many further developments both in the temporal and the spatial domains.

2D-PIR representation [Nabi95] is another approach that adapts two existing representation formalisms (Allen's temporal intervals and 2D-strings) and combines them in a novel way to produce a unified representation. The basic idea of 2D-PIR is that each spatial object is projected along the x and y axes forming an x-interval and y-interval for the object. A 2D-PIR is defined as a triple (δ, χ, ψ) where δ is a topological relation from the set {disjoint, meets, contains, insides, overlaps, covers, equal, and covered-by}. χ and ψ are interval spatial relations {before, equal, meets, overlaps, during, start, finish, after, meet-inverse, overlap-inverse, during-inverse, start-inverse, and finish-inverse}. χ represents the interval spatial relation along the x-axis, and ψ represents the interval spatial relation along the y-axis. Then a picture can be represented by a 2D-PIR graph which is a connected labeled digraph G(V, R) where V is a finite nonempty set of symbols representing objects in the picture, and R is a set of edges labeled by 2D-PIR spatial relations.

The 2D-PIR is simple and powerful. However, the information gathered from the intervals on the X axis and Y axis are sometime meaningless and redundant when it is combined with the topological relations. For example, one or both intervals (on X or Y axes) overlapped do not mean objects overlapped. In Figure 2.15(a), although the intervals are overlapped, the objects are not. Therefore the directional part of the 2D-PIR is meaningless in this case. However, if one of the intervals is disjoint, this means that the objects are disjoint. In Figure 2.15(b), the topological part of the 2D-PIR is redundant since it can be concluded from the fact that the X intervals are disjoint.



(a) Intervals are overlapped whereas objects are disjoint



(b) X intervals are disjoint the and so objects are disjoint

Figure 2.15. Some deficiencies with 2D-PIR representation

2.6.3 Symbolic array

Basically, symbolic arrays can be regarded as multidimensional sparse indexes that allow nested representation, namely, the element of the array can be in turn a symbolic array [Papa94]. The 2D array structures are used to represent a set of spatial relations between representative points. These points are placed in the proper cells in the spatial index and are thus different from the use of projections. Spatial indexes can also be used to represent directional relations using either one or four points per object, and topological relations by analyzing the indexes stored in the cells. In fact, symbolic array is very similar to the 2D-string method except it allows hierarchical representation that is useful to represent complex spatial structure in different levels. It is, however, based on the assumption that the positions of the spatial entity are known. While it is quite straightforward for directional relations, like 2D string and its variants, it does not provide any way to accommodate topological relations such as overlap, contains etc.

2.6.4 Polar coordinates system

Making a picture representation that is rotation-invariant becomes a hot topic recently. This is witnessed by the proposals of 2D R-string [Petr93], [Petr94] and RS-string [Huan96]. They all make use of polar coordinates system based on a designated center (or origin).

A 2D R-string [Petr93] is obtained from a cutting mechanism that generates cutting concentric circles along the ring-direction and cutting radial segments along the sector-direction based on some rotation center. This rotation center will be treated as the origin of a polar coordinates system. The beginning and ending points of an object in the sector-direction are defined by the two radial segments tangent to the object while the beginning and ending points of an object in the ring-direction are defined by the two concentric circles tangent to the object. The string generated along the ring-direction is called a C-string while the string generated along the sector-direction is called a S-string. A 2D R-string is an ordered pair of C-string and S-string. However, the rotation center object is always missing from a 2D R-string representation for a picture. Thus it loses the information about the spatial relation between the rotation center object and other objects. Although 2D RS-string [Huan96] improves the 2D R-string, the main problem about how to find an appropriate polar center remains unsolved. Moreover, the object chosen as the polar center may not appear in all database pictures. This means that the reference frame is changed from picture to picture, and therefore the method is not applicable.

2.6.5 Other representations

Besides the approaches we mentioned above, there are many other representations for spatial knowledge.

Forbus, Nielsen, and Faltings [Forb91] present a framework for qualitative kinematics (geometric aspects of representing and reasoning about motion) based on combining metric diagrams with a qualitative place vocabulary. They claim that there is no powerful, general purpose, purely qualitative spatial representation (poverty conjecture), and thus introduce the place vocabulary as a symbolic description of shape and space, and state transitions.

Chandrasekaran and Narayanan [Chan90] propose a hybrid system intended to integrate "modality-specific" (such as visual) mechanisms with abstract symbolic representations. The "Image Representation System" (IRS) is made out of "Image Symbol Structure" (ISS). ISS are hierarchical discrete symbol representations consisting of parameterized primitives, which encode only the intrinsically visual aspects of a scene. Among those primitives are the “symbolic percepts” (S-percepts) containing descriptions of spatial relations among the volumetric parts of an object. The corresponding “analogical percepts” (A-percepts) are the mental pictures that result from the interpretation of S-percepts.

Huang and Jean [Huan95] introduce an approach to describe the knowledge structures of pictures based on morphological skeleton representation. The skeleton SK(X) of an object X is the set of centers of the maximal disks inscribable inside X. SK(X) can be divided into subsets whose members having the same radius in their corresponding maximal disks. Such a subset of SK(X) is called a skeleton subset. The approach is to encode a picture as a set of object skeletons incorporated into a picture knowledge structure called the SK-set. Then a sketch can be partially reconstructed and progressively refined back to the original picture by dilating the skeleton subsets in a SK-set. This approach is more accurate for representing object spatial relations. However, it needs more storage than other representations.

Though a tremendous amount of work [Papa98], [Tani98], and [Huan04] has been done on the representation of spatial knowledge and its retrieval, we only manage to touch upon some typical approaches to demonstrate the weaknesses of existing approaches. In the following, we will discuss some systematic spatial relation representations and introduce some novel similarity measure and index for automatic similarity retrieval.

Chapter 3

Systematic spatial relation representation IO&T

3.1 Two classes of spatial relations

In previous chapters, we introduced some early 2D spatial relation representation methods that are based on the concept of 1D interval spatial relation. Although they have over simplified the 2D spatial relations, they still can be used to solve some practical problems. However, these methods fail to evolve further in building a general representation model due to their inherent deficiency. Therefore, most recent researches have turned to the study of spatial relations that are more relevant to objects in 2D space.

In spatial relation, two factors determine the relative position of objects in 2D space: the relative orientation of objects to each other, and their extensions. Considering these factors independently from each other results in two classes of spatial relations: the topological relations (ignoring orientation) and the orientation relation (ignoring extension by regarding objects as points). Topological relations describe how the boundaries of the two objects relate, and orientation relations describe where the objects are placed relative to one another. These two classes of spatial relations have been studied individually in some specific application areas. Although 2D string and other early works have combined these two classes heuristically, there should be a more systematic approach to combine the two kinds of spatial relation into a unified framework ([Hern94] and [Nabi96]) so that a similarity measure can be defined based on it. Before we combine topological and orientation spatial relation, we need to analyze and define a small set of spatial relations from the relevant topology and orientation more formally. In the following, we will first discuss the topological and orientation relation individually, and then propose a new representation to combine them.

3.2 Topological relations

If we disregard the relative orientation of objects to each other, we are still able to describe the topological relation between two objects based on the following:

- they are far from each other,
- they are close, but do not touch,
- they touch,
- they overlap, or
- one of them is included in the other.

This seems to be a reasonable set of topological relations to start with though it turns out to have several drawbacks. The first distinction between objects close together or far apart cannot be defined by topological means. It requires an external frame of reference with respect to which the positions of the objects being compared must be established first. We shall see later that the relative size of object provides one such external criterion allowing a coarse form of far/close distinction. For the time being, however, we rule out distance since it is not preserved by topological equivalence. We only concern about the completeness of the set of distinctions as to whether it covers all possible situations. Also, the corresponding topological relations should be mutually exclusive, so that each possible situation corresponds to one and only one topological relation. The given set of topological relations does not fulfill these requirements.

Set theory is a good starting point to describe topological relations: The objects are given by the sets of points they consist of, and the topological relations are defined in terms of set operations. Guting [Guti88], for example, gave the following definitions in terms of the set operations =, ≠,⊆, ∩:

$$x = y \quad := \text{points}(x) = \text{points}(y)$$

$$x \neq y \quad := \text{points}(x) \neq \text{points}(y)$$

$$x \text{ inside } y \quad := \text{points}(x) \subseteq \text{points}(y)$$

$$x \text{ outside } y \quad := \text{points}(x) \cap \text{points}(y) = \varnothing$$

$$x \text{ intersects } y \quad := \text{points}(x) \cap \text{points}(y) \neq \varnothing$$

As Egenhofer and Franzosa [Egen91] pointed out, this set of topological relations is neither complete nor mutually exclusive. As defined, equal and inside are special cases of intersects. There is also no way of distinguishing touches from intersects (overlaps), because that distinction is based on the difference between the boundary and the interior of point sets, which is not defined here.

That observation led Egenhofer and Franzosa [Egen91] to the definition of topological relations based on the four intersections of the boundaries (∂) and interiors(°) of two sets A and B. The interior of a set A is the union of all open sets in A. The boundary of a set A is the intersection of the closure of A and the closure of the complement of A. The closure of A is the intersections of all closed sets containing A. The complement of A with respect to the embedding space is the set of all points of embedding space not contained in A. Since an intersection can be either empty (∅) or not empty (¬∅), for the four intersections $\partial A \cap \partial B$, $A° \cap B°$, $\partial A \cap B°$, $A° \cap \partial B$ we obtain $2^4=16$ topological relations (Table 3.1). Figure 3.1 shows geometrical interpretations of these 16 topological relations. Obviously, some of these topological relations do not have meaningful physical interpretations. For example, r_2 implies projections without boundaries, r_9 requires one of the objects to have a boundary but no interior, and so on. Imposing the restrictions of physical space, we can in fact eliminate half of them (Table 3.2). The remaining eight topological relations (disjoint(dt), touch(to), overlap(ov), covers(co), inside(in), coverby(cb), contain(ct)s, and equal(eq)) fulfill the criteria mentioned above: They are mutually exclusive and all possible situations allowed by the assumptions can be described by one of them. Table 3.2 shows their specification in terms of the intersection of boundaries and interiors.

Table 3.1. The 16 possible topological relations

XφY	∂∩∂	°∩°	∂∩°	°∩∂
r_0	∅	∅	∅	∅
r_1	¬∅	∅	∅	∅
r_2	∅	¬∅	∅	∅
r_3	¬∅	¬∅	∅	∅
r_4	∅	∅	¬∅	∅
r_5	¬∅	∅	¬∅	∅
r_6	∅	¬∅	¬∅	∅
r_7	¬∅	¬∅	¬∅	∅
r_8	∅	∅	∅	¬∅
r_9	¬∅	∅	∅	¬∅
r_{10}	∅	¬∅	∅	¬∅
r_{11}	¬∅	¬∅	∅	¬∅
r_{12}	∅	∅	¬∅	¬∅
r_{13}	¬∅	∅	¬∅	¬∅
r_{14}	∅	¬∅	¬∅	¬∅
r_{15}	¬∅	¬∅	¬∅	¬∅

Table 3.2. Specification of binary topological relations

XφY		∂∩∂	°∩°	∂∩°	°∩∂
r_0	disjoint (dt)	∅	∅	∅	∅
r_1	touch (to)	¬∅	∅	∅	∅
r_3	equal (eq)	¬∅	¬∅	∅	∅
r_6	inside (in)	∅	¬∅	¬∅	∅
r_7	coverby (cb)	¬∅	¬∅	¬∅	∅
r_{10}	contains (ct)	∅	¬∅	∅	¬∅
r_{11}	covers(co)	¬∅	¬∅	∅	¬∅
r_{15}	overlaps (ov)	¬∅	¬∅	¬∅	¬∅

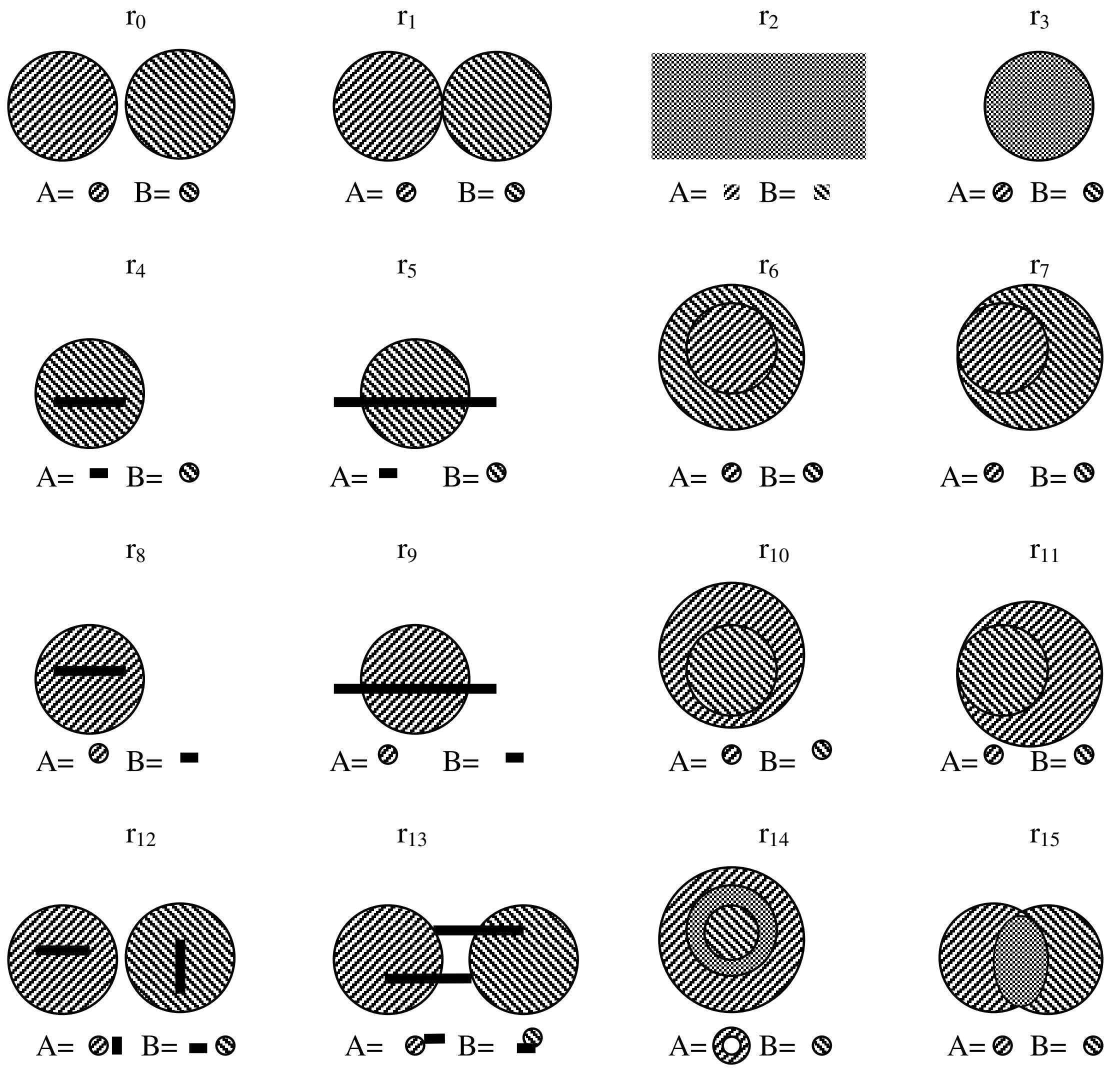


Figure 3.1. Geometrical interpretations of binary topological relations

3.3 Orientation Relations

Orientation relations describe where objects are placed relative to one another. Three elements are needed to establish an orientation: a primary object, a reference object and a frame of reference. In the following, we will derive sets of orientation relations at various granularity levels.

Considering only points, the first level of orientation relations can be constructed by connecting the point of view and the reference object by a straight line such that the primary object can be to the left, to the right, or on that line. A complementary set of basic orientations results from the perpendicular line dividing the plane at the reference point. Here again, the primary object can be above, below or on the dividing line. The resulting orientation relations are called back, front, and collinear respectively.

The two basic sets of orientation can be combined to obtain the next level of finer distinctions (Figure 3.2). Here again, two sets of orientation relations are possible. One set results from the superposition of the basic sets of the orientation relations to obtain left-back, left-front, right-front, and right-back. The other set that is more commonly used requires a rotation of the axes by 45 degrees in addition to the superposition and contains the orientation relations back, left, front, and right. Co-linearity, which plays a key role on the first level to handle the case in which no distinction can be made between left/right or back/front, is represented only implicitly at level 2. The assumption is that the pairs of orientation relations left/right and back/front complement each other, such that if no distinction is possible between left and right, then it must be back or front, and vice versa.

The next level with the eight distinctions front, back, left, right, left-back, right-back, left-front, and right-front is built in a similar manner by superimposing and rotating the axes from the previous level. At each level, the sets of orientation relations given above are complete and their elements are mutually exclusive. A solution proposed by Hogg and Schwarzer [Hogg91] introduces a subdivision in 16 sectors, which is claimed to be fine enough to express any of the orientation relations of the 3 levels considered here as a range of sectors (Figure 3.2).

However, qualitative representation of orientation is a fuzzy concept that largely depends on the size of the objects and the distance between them. To facilitate the understanding of the proposed orientation category, we use the second representation of level 3 (Figure 3.2(7)) in the following discussion, and substitute different ranges with 8 orientation codes {f, rf, r, rb, b, lb, l, lf} which represent front, right-front, right, right-back, back, left-back, left, and left-front respectively as shown in Figure 3.3(b). It is clear that every sector in the common level (Figure 3.2(1)) is included in only one orientation sector in Figure 3.2(7).

With the above categorization, the problem of sudden change in orientation when crossing the boundaries between neighboring regions arises. As we see, sector 1 and sector 0 in the common level (Figure 3.2(1)) are categorized in Figure 3.2(7) to two different orientation sectors that are called rf and r respectively in Figure 3.3(b). However, based on our intuition, sector 1 can be argued to be categorized to r sometime, and on the other hand, sector 0 may be categorized to rf according to different requirements. There is no good reason why sector 1 and sector 2 must be categorized to the same orientation sector whereas sector 1 and sector 0 must be categorized to two different orientation sectors even though the angle differences between sector 1 with sector 2 and sector 0 are the same. Using higher level of orientation does not solve the problem.

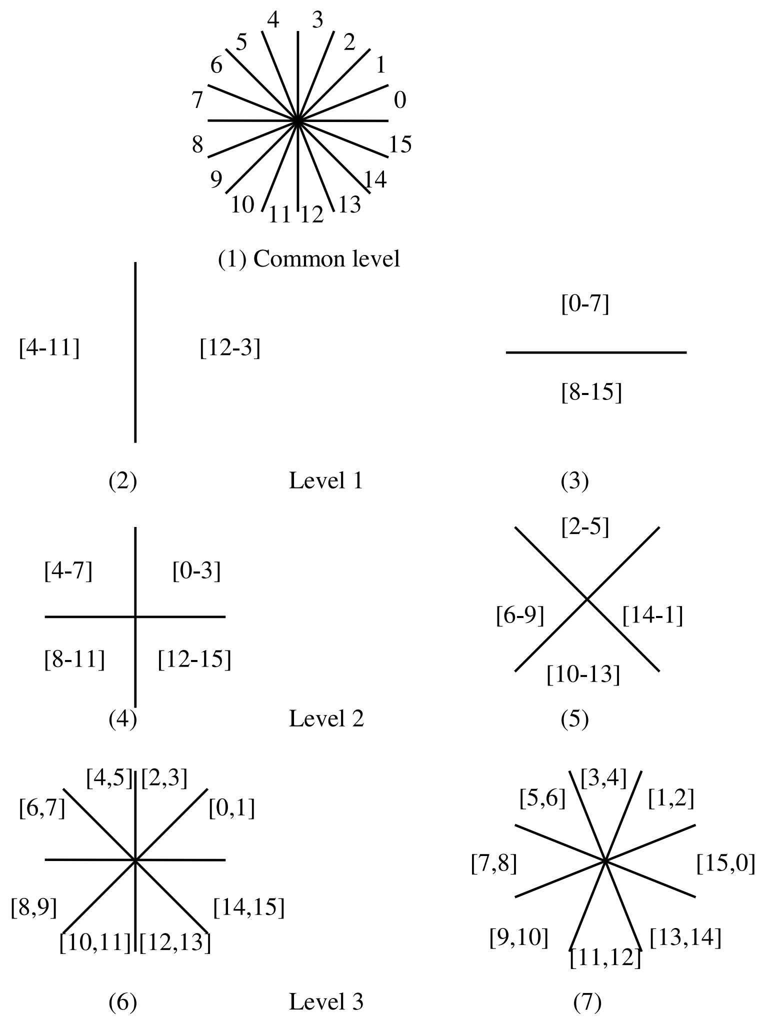
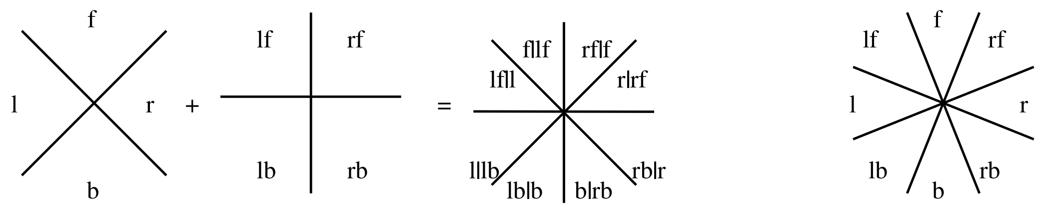


Figure 3.2. Range representation of orientations (adapted from [Hogg91])

3.4 Fuzzy orientation

In order to solve the above problem, a fuzziness concept was introduced [Ang99] to the normal eight orientation relations in Figure 3.3(b) to allow the categories of orientation relations to be overlapped. Given two neighboring orientation relations R and S, the overlap part between them is denoted as R|S. Interpreting the overlap part R|S as either sector R or sector S are both acceptable. Using this notation, the new sector codes are r|rf, rf|f, f|lf, lf|l, l|lb, lb|b, b|rb, rb|r. Figure 3.3(a) shows clearly the effect of fuzziness to each of the 8 normal orientations and how the overlapping induces the eight fuzzy orientation relations.



(a) Eight fuzzy orientation relations (b) Eight normal orientation relations

Figure 3.3. The fuzzy category

According to the new fuzzy category, we have three ways to interpret an orientation. For example, the orientation rlrf can be taken as the sector itself, or it can also be treated as r or rf. Which of the three interpretations is used is application dependent. For example, in Figure 3.4, there are four objects A, B, C, and D. The center points of B, C, and D are all in sector rlrf with respect to the center point of A, but the size and shape of B, C, and D are different. According to our common perception, C is usually categorized to orientation relation code r, and D is categorized to rf. As for B, it is so fuzzy that we will use rlrf instead.

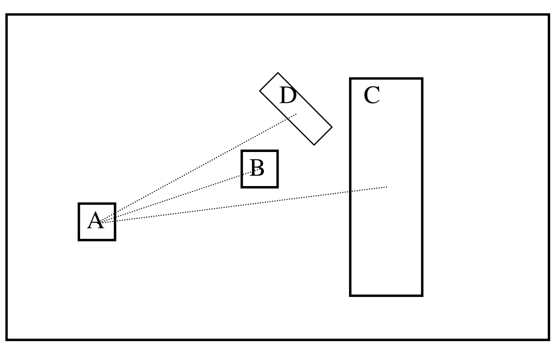


Figure 3.4. The orientation spatial relations of B, C, and D with respect to A

With the use of the fuzzy category in describing the orientation relation, we overcome the problem of sudden change in the orientation of the objects to be described. We endow an application the flexibility to interpret the orientation codes according to its need. When an object is found to be in a sector without ambiguity, the respective normal orientation code is used. Otherwise, we represent it by a fuzzy code. Since we allow ambiguous code to represent a spatial relation, the representation itself looks fuzzier. However, this fuzzy representation will benefit the similarity retrieval in the following discussion.

3.5 Combining topological and orientation relation - IO&T representation

IO&T is a representation scheme that combines fuzzy Intrinsic Orientation and Topological spatial relation. As the name suggests, the discussion for IO&T representation should start from intrinsic orientation.

3.5.1 Intrinsic orientation relations

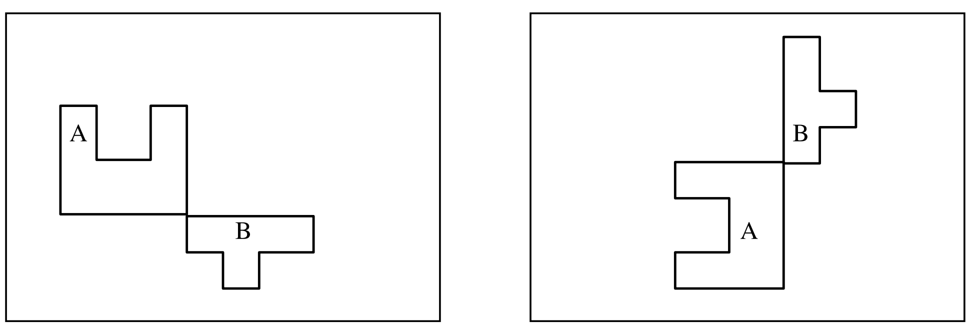
The orientation that determines the direction in which the primary object is located in relation to the reference object must be given with respect to a reference frame. In the sentence "The ball is in front of the car", the reference frame is based on the car and its front is clearly defined. There are three types of orientations:

- Intrinsic orientation: The orientation is given by some inherent property of the reference object (For example, with respect to the car front, the ball is in front of the car). Some criteria used to determine the intrinsic orientation of objects and places are: the direction of motion of the object, the side containing perceptual apparatus, the side oriented towards the observer, the symmetry of objects, etc.
- Extrinsic orientation: The orientation on the reference object is imposed by external factors. Relevant factors are the accessibility of the reference object, its

motion (or that of the observer), or other objects in its vicinity or the earth gravitation. If a car is moving backwards, that direction is considered “front”.

• Deictic Orientation: The orientation is imposed by the point of view from which the reference object is seen by an observer within the scene or from the speaker’s point of view.

Normally, the reference frame with respect to which the orientation is determined can be the combination of the above three different types. Among these three reference frames, the intrinsic orientation is the most natural frame of reference for the relative orientations among objects (The intrinsic orientation has been discussed and analyzed in cognition field in detail such as [Dilw02]). By using the intrinsic orientation, we have no need to worry about the rotation of the pictures. For example, the picture in Figure 3.5(b) is a rotated version of the picture in Figure 3.5(a). If we use extrinsic or deictic reference frame, their representations are totally different. However, by using the intrinsic orientation, there will be no difference between Figure 3.5(a) and 3.5(b).



(a) (b)

Figure 3.5. Two example pictures ((b) is a rotated version of (a))

Therefore, we will use intrinsic orientation in our following discussion. Before that, we define intrinsic orientation formally as:

Definition 3.1. An *intrinsic orientation* of an object in a picture is the directional relation between another object in the same picture and this object based on the nature and property of the object itself.

In other words, the reference frame is embedded into the object itself. It is independent of the point of view completely.

3.5.2 IO&T graph representation

Based on the conclusion of the above analysis, we proposed to use a tuple **<A, [TR, OR1, OR2], B>** to represent the spatial relation between two objects A and B where TR is the topological relation between object A and object B, OR1 represents the orientation relation of A to B based on A's intrinsic orientation, and OR2 represents the orientation relation of B to A based on B's intrinsic orientation.

Since the IO&T representation involves two objects, a picture with n objects could be represented by n*(n-1)/2 tuples $<O_i, [Rt_{ij}, Ro1_{ij}, Ro2_{ij}], O_j>$ where O_i, O_j are objects and $[Rt_{ij}, Ro1_{ij}, Ro2_{ij}]$ are spatial relations between O_i and O_j. We may order the name of objects in all tuples in order to uniquely represent a picture by IO&T tuples. The following is the IO&T tuples construction algorithm.

Algorithm 3.1: IO&T tuples construction

Input: A preprocessed picture with n objects (with their intrinsic orientations)

Output: A set of tuples R using IO&T representing the picture

begin

1. *Order objects alphabetically according to their name.*
   Let V be the ordered set of objects
2. *for (i=1, n-1, i++){ /* n is the number of objects in the picture */*
3. *for (j= i+1, n, j++){*

4. *Compute* Rt_{ij} */** Rt_{ij} *is the topological relation between V[i] and V[j] */*
5. *Compute* $Ro1_{ij}$ */** $Ro1_{ij}$ *is the orientation relation between V[i] and V[j] based on the intrinsic orientation of V[i] */*
6. *Compute* $Ro2_{ij}$ */** $Ro2_{ij}$ *is the orientation relation between V[i] and V[j] based on the intrinsic orientation of V[j] */*
7. *put* $<V[i], [Rt_{ij}, Ro1_{ij}, Ro2_{ij}], V[j]>$ *into R*
8. *}*
9. *}*
10. *return R*

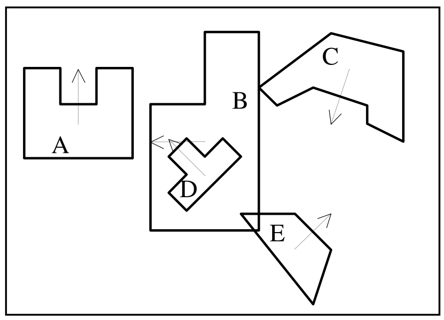
end

A picture can be represented as G(V, R) where V is a finite nonempty set of symbols representing the objects in the picture and R is a set of IO&T tuples, and G(V, R) can be implemented as an adjacency list. It should be noted that a little deformation (such as rotate the whole picture or enlarge etc) of a picture would not affect the structure of its adjacency list because the IO&T tuples are organized alphabetically. Two pictures with the same objects will always have the same structure of their adjacency lists. This is an important feature that is very useful for picture retrieval.

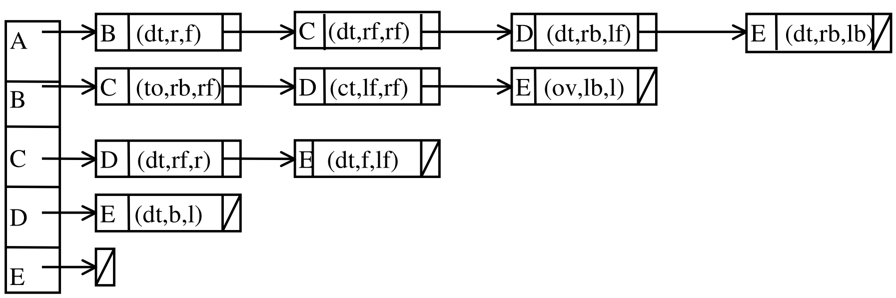
We use the sample picture P in Figure 3.6(a) to see how the IO&T representation is generated for this picture. The V of picture P is {A, B, D, E, C}. All arrows in the picture represent the intrinsic fronts of the related objects. After step 1 of algorithm 3.1, the V will become {A, B, C, D, E}. Following steps 4, 5, and 6, since A and B are disjoint, the geometric center of B is on the right of A, and the geometric center of A is in the front of B, we get <A, [dt, r, f], B>, the first IO&T tuple in R. Other tuples generated are <A, [dt, rf, rf], C>, <A, [dt, rb, lf], D>, <A, [dt, rb, lb], E>, <B, [to, rb, rf], C>, <B, [ct, lf, rf], D>, <B, [ov, lb, l], E>, <C, [dt, rf, r], D>, <C, [dt, f, lf], E>, and <D, [dt, b, l], E>. The corresponding adjacency list is shown in Figure

3.6(b). Note that the name of the first object in a tuple is the index to the array of list heads, and the name of the second object in the tuple is stored with their spatial relation in a node of the linked list according to its alphabetical order.

Compared with the tuple **<A, [Rt, Ro], B>** used in [Hern94], the IO&T tuple has one more element. However, we are able to represent the mutual orientation relations between two objects within a single tuple, and pictures can be more accurately represented with slightly higher space requirement.



(a) A picture P (All arrows represent intrinsic fronts of objects)



(b) The corresponding adjacency list of picture P

Figure 3.6. A picture P and its corresponding adjacency list

3.6 Compare with other systematic representations

[Hern94] is one of typical approach using systematic spatial relation representation. Hernandez proposed to describe the spatial relation between two objects with the following tuple in which both the topological and the orientation relations are used:

<primary_object, [topological , orientation], ref_object, ref_frame>

where primary_object is the object to be located, ref_object is the object with which the primary_object is to be located and the ref_frame is the reference frame that is used when representing the spatial relations between the two objects.

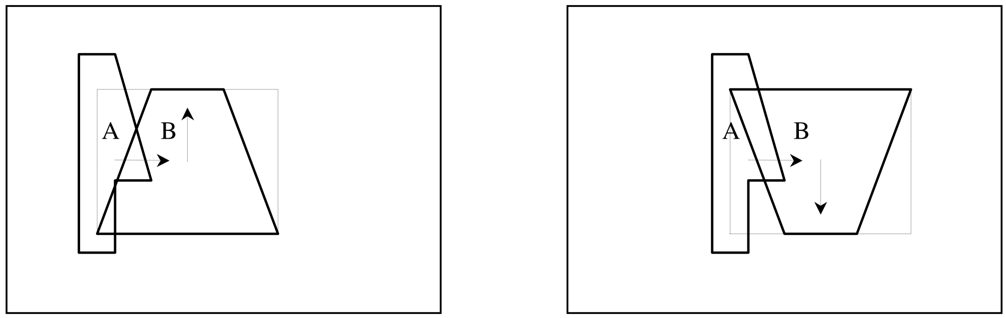
This representation has some advantages:

1. Topological and orientation relations combined will describe spatial relation in 2D space clearer.
2. The combination of the topological and orientation relations is flexible. They can be handled individually or as a unified spatial relation. It is application oriented.
3. There are many combinations of the two component spatial relations. We don't have to name every possible combination artificially. The interpretation of the spatial relation is more straightforward.

However, there are two major drawbacks in [Hern94]'s approach, which make big difference between IO&T and [Hern94].

First, [Hern94] is not as good as IO&T in the accuracy of the representation:

The problem lies in its orientation relation in which only one of the two objects is used as the reference in describing the orientation relation. It assumes the orientation from object A to object B is always complement to that from B to A. Therefore, only one orientation relation is needed and the other can be inferred. This is not always true especially when the rotation information of objects is to be considered. For example in Figure 3.7, there is no difference between picture (a) and picture (b) when object A is chosen as the reference object to describe the orientation of object B. These pictures are clearly different when B is the reference object. If every object faces different direction, then we have to use two Hernandez's tuples to represent the orientation relation between two objects. However, regardless which reference frame is used, there is only one topological relation between two objects. Therefore, the topological relation parts of these two tuples are identical and one of them is redundant. On contrast, this problem is resolved in IO&T representation.



(a) (b)

Figure 3.7. No difference between the representation of (a) and (b) when A is the reference object (All dotted arrows represent the fronts of objects)

In fact, IO&T has implicit reference frame by using intrinsic orientation (i.e. the reference frame is the same as the primary object itself if we use [Hern94]'s concept. This is a big saving from representation perspective.

Second, [Hern94] has the weakness in the feasibility of similarity measure:

Hernandez's approach needs an explicit reference frame. There are two options here:

- Suppose the reference is based on a point of view, the representation will not be invariant under transformation, which is a very desirable feature for picture representation.
- Suppose the reference is based on one fixed object inside the picture, this should have no problem for the representation. However, the representation is for the purpose of similarity retrieval in our following discussion. If the fixed object is not in both query picture and database picture, we will have problem to compare the spatial relations between pictures. It is unlikely that we can find such a reference object appearing in all pictures.

Actually, [Hern94]'s work focus more on representation for spatial database (i.e. spatial reasoning in spatial database). There is no similarity measure proposed for similarity retrieval in his work, which, however, is one of our major objectives in this thesis. This point will be clearer in the following discussion.

Besides Hernandez's representation, another typical systematic representation is 2D-PIR [Nabi95]. Since we have already some discussion about the representation of 2D-PIR in section 2.6.2, we will not repeat here in detail. In short, 2D-PIR has some duplicated information captured in its topological representation part and directional representation part. 2D-PIR is not invariant under rotation, and it requires some complicated transformation for similarity comparing. Since 2D-PIR has a similarity measure proposed for similarity retrieval, we will compare IO&T with 2D-PIR experimentally for its similarity retrieval in the next chapter.

3.7 Summary

In this chapter, we have discussed topological relation and orientation relation individually and theoretically. A spatial relation representation IO&T is proposed to combine topological relation and orientation relation more systematically. Comparing to Hernandez's representation, the spatial characteristics of objects can be described by IO&T more accurately and naturally. In particular, IO&T is proposed for similarity retrieval instead of spatial reasoning. IO&T overcomes the ambiguity problems that exist in many other representations. The advantages of IO&T representation are

- Invariant under the translation and rotation of a picture;
- Flexible and comprehensive for the topological and orientation representation;
- Ease in defining a similarity measure.

The last point is critical because similarity retrieval is more applicable in many applications. In the following chapter, we will discuss how we are going to design a metric system to measure the similarity based on the IO&T representation.

Chapter 4

Similarity measure based on spatial relation representation

In previous chapters, we have discussed iconic spatial relation representations. IO&T representation has been proposed. However, the objective of representation is for the ease of retrieval, i.e. how to retrieve the needed iconic picture from the database. This requires a similarity measure for comparing the similarity based on the representation. Pictorial similarity measure is different from the text-based similarity measure. In this chapter, we will discuss the similarity measure in detail for iconic spatial relation based retrieval. We will start our discussion about the concept of similarity measure for picture retrieval. Some native similarity measures will be used to describe the different aspects of pictorial similarity measures. Based on the analysis of these native approaches, we will introduce a simulated measure and a degree based measure using IO&T as the representation template. We will also highlight an object-based similarity concept on top of the proposed similarity measure to integrate the advantages of proposed measures.

4.1 What is similarity measure

There are two kinds of picture retrieval: *exact matching* and *similarity retrieval*, with the former a special case of the latter. *Exact matching* is to retrieve from the picture database those pictures that are exactly the same as a query picture or contain the query picture. S*imilarity retrieval* is to retrieve those pictures that are similar to the query picture given by a user. In similarity retrieval, the retrieved pictures can either be the same as, contain the query picture, or they can have many similar parts with the query picture. For example, there are four pictures P1, P2, P3, and P4 in Figure 4.1. P1 is a query picture, and P2, P3, and P4 are database pictures. P1 is neither completely the same as P2, P3, or P4, nor a sub-picture of anyone of them. To find the picture that is most similar to P1, we must design a real-valued function that can measure the similarity of every pair of pictures. This function is called s*imilarity measure*.

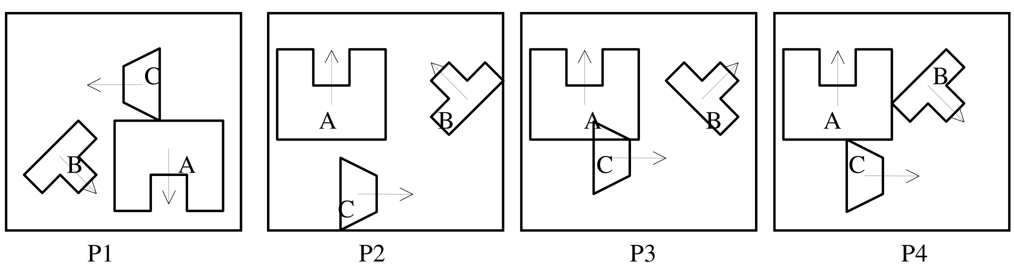


Figure 4.1. Example pictures (All arrows represent intrinsic fronts of objects)

In the following, we will discuss some similarity measures. The discussion will start from two simple native similarity measures and derive two similarity measures based on the native measures. An object-based similarity measure is proposed to offer the advantages from the other measures. Some experiments have been done to compare the effectiveness of proposed similarity measures.

We assume no duplicated object in a picture as it has been proved in [Tucc91] that the similarity retrieval problem is NP-hard when multiplicity of objects is allowed in picture matching.

4.2 Native similarity measures

A similarity measure is used to measure the similarity between pictures. However, unlike exact matching, similarity is a subjective concept. In this section, we will discuss two straightforward but intuitive similarity measures to demonstrate the differences of similarity measures in various aspects.

4.2.1 Common clique similarity measure

Since iconic spatial relation representation is based on spatial relations between objects to represent a picture, one way to measure the similarity between two iconic spatial relation representations is based on the size of the clique of objects with matching spatial relations. The detail of making use of common clique concept is illustrated as follows:

As the first step of comparing two pictures, we identify objects having the "same position". The concept of "same position" does not require the objects to appear at the same place, or to be static with respect to each other. They will be treated as "in the same position" so long as the objects' movements do not change the spatial relations among themselves. We are going to find between two pictures the maximal common sub-picture in which not only the objects are the same, but also the spatial relations between any two objects in the sub-picture are the same. If we represent a sub-picture by a graph in which each node represents an object and the line connecting two nodes is labeled with the relation between corresponding objects, then the maximal common sub-picture forms a clique.

In the following discussion, we use IO&T for pictures representation. Given two pictures represented by (V1, R1) and (V2, R2) respectively where V1 and V2 are sets of objects, and R1 and R2 are sets of IO&T spatial relation tuples, the following algorithm can be used to find the maximal common sub-picture between them.

Algorithm 4.1: Finding the maximal common sub-picture

Input: (V1, R1) and (V2, R2), the IO&T representations of two pictures
Output: the maximal common sub-picture (V3,R3)
begin

1. *for each Oi in V1 if there is a Oj in V2 such that Oj= Oi*
2. *then record Oi as a vertex*
3. *endfor*
4. *for each pair of vertices Om, On*
5. *if* $<O_m, [Rt_{mn}, Ro1_{mn}, Ro2_{mn}], O_n>$ *in R1 and R2 are the same then construct an edge between them and labeled with* $[Rt_{mn}, Ro1_{mn}, Ro2_{mn}]$
6. *endfor*
7. *find the maximal complete sub-graphs (cliques) of this undirected graph, record all vertices in the clique in V3 and the its corresponding spatial relation in R3*
8. *output (V3, R3)*

end.

The algorithm for generating the maximal complete sub-graphs (cliques) developed by [Bron73] is shown below. It is expressed as a recursive program.

Algorithm 4.2: Generating the clique

Input: An undirected graph G=(V, E), where v is the set of vertices, E is the set of edges

Output: A set of S vertices constituting a clique

```
begin
    initialize S as an empty set
    call subroutine clique (V, ∅)
      subroutine clique(N, D)
       begin
          if N∪D=∅
           then output S, which is a clique
          else
             if N ≠∅
             then select one vertex f from N
                call subroutine explore(f)
                while N∩(V-Adj(f)) ≠∅ do
                 v is a vertex in N∩(V-Adj(f))
                 call subroutine explore(v)
               endwhile
            endif
        endif
    end-of subtoutine-clique
subroutine explore(u)
begin
     let N be N-{u}
    let S be S∪{u}
    call subroutine clique(N∩Adj(u), D∩Adj(u))
    let S be S-{u}
    let D be D∪{u}
    end-of-subroutine-explore
end
```

After using the above algorithms to find the maximal sub-picture, we may treat the common sub-picture as a fixed reference part in which all objects remain at their old positions in both pictures. It is possible that the maximal sub-picture is not unique. However, what we need to find out is the number of objects inside the maximal common clique. This number should be unique regardless if there is a unique maximal common clique. The similarity measure of common clique approach, $\mathbf{D_{clique}}$, is defined as:

$$\mathbf{D_{clique} = N_q - N_{max\_clique}} \qquad \textbf{(4.1)}$$

Where N_q is the number of objects in the query picture. N_{max_clique} is the number of objects in the common clique.

For a database picture to be the most similar to the query picture Q, the number of objects that have changed their spatial relations must be the smallest, i.e. D_{clique} is smallest.

However, the above measure of similarity is only of theoretical interest. It has over-simplified the similarity retrieval problem into finding the largest common clique in the corresponding two graphs. If there is only one object in the clique found by the above algorithm, then it is hard to agree that all pictures containing the objects in the query picture to be considered similar. In addition, the common clique algorithm is NP-hard and hence it is not very practical to use it to solve the problem. Therefore, in the following, we introduce another similarity measure which uses the number of pair-wise spatial relation changes as the measure of similarity.

4.2.2 Spatial relation measure

Basically, the spatial relation measure is based on pair-wise spatial relations. Given a query picture with n objects, we need to compare each of the query picture's n*(n-1)/2 spatial relation tuples with that of each database picture, and the number of same spatial relation tuples (i.e. two objects are the same, and the spatial relations including topological relation part and orientation relation part are the same) is used as the similarity measure.

Definition 4.1. The spatial relation measure $\mathbf{D_{srm}(Q, P)}$ between a query picture Q and a database picture P is defined as the number of spatial relation tuple $<O_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], O_j>$ in Q such that there is corresponding spatial relation tuple $<O_m,[TR_{mn}, OR1_{mn}, OR2_{mn}],O_n>$ in P where $O_m = O_i$, and $O_n = O_j$ and $TR_{ij} = TR_{mn}$ and $OR1_{ij} = OR1_{mn}$ and $OR2_{ij} = OR2_{mn}$ (where O_i and O_j are two different objects in Q, and TR_{ij}, $OR1_{ij}$ and $OR2_{ij}$ are topological relation and two orientation relations respectively based on IO&T; O_m, O_n, TR_{mn}, $OR1_{mn}$, and $OR2_{mn}$ are referring to the related components in P).

The algorithm to find the matching tuples as shown in Algorithm 4.3 is quite straightforward.

Algorithm 4.3: The number of matched tuples

Input: (Vq, Rq) and (Vp, Rp), the IO&T representations of the query picture Q and a database picture P.

Output: D(Q, P), the number of same tuple spatial relations between Q and P.

begin

1. *D=0*
2. *if* $Vq \subseteq Vp$ *then*

3. *for each* $T1 = \langle O_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], O_j \rangle$ *in Rq*

4. *find* $T2 = \langle O_m,[TR_{mn}, OR1_{mn}, OR2_{mn}],O_n \rangle$ *in Rp*

   such that $O_m = O_i$, *and* $O_n = O_j$ *and* $TR_{ij} = TR_{mn}$ *and* $OR1_{ij} = OR1_{mn}$ *and* $OR2_{ij} = OR2_{mn}$

   add 1 to D

5. *endfor*

6. *elseif set D to -1*

7. *endif*

8. *return D*

end

The return value D of Algorithm 4.3 is used as the similarity measure D_{srm} directly. If D is bigger, the database picture is more similar to the query picture. The value -1 means that the database picture does not have all the objects in the query picture. Therefore, the pictures are not similar.

Comparing with the common clique similarity measure, this measure is more practical as its complexity is only $O(n^2)$. However, this measure is inadequate because the degree of spatial relation changes that also affect human similarity perception has not been considered.

4.3 Derived similarity measures

The two native similarity measures discussed in previous session are simple and intuitive. However, they may not be very practical in a real similarity retrieval system either due to the algorithm complexity or over simplified of the concept of similarity of human being. Therefore, in this section, we propose two derived similarity measures to make up the disadvantage of the native measures.

4.3.1 Degree similarity retrieval

Degree similarity retrieval is to retrieve pictures according to the degree of spatial relations changes. To quantify the concept "degree of similarity", we must analyze the structure of topological and orientation domain first.

4.3.1.1 Structure of the topological domain

The structure of the topological relational domain is derived from the structure of physical space. For topological domain, the topological relations are described in Table 3.2 and we may construct the neighborhood structure involving these topological relations as in Figure 4.2. Two topological relations are neighbors if one can be transformed directly into another by scaling, moving, or rotating an object. Thus, disjoint (dt) and touch (to) are neighboring topological relations, because two disjoint objects can come closer and touch each other (touch) without going through an intervening state. Overlap (ov) and contain (ct) are two topological relations that are not neighbors, because one object containing another can touch it before overlap it. The notion of topological neighbor is described as follow:

Definition 4.2. Two topological relations are *neighbors* if one of the relation can be directly transformed into the other without going through any other topological relation by continuously deforming (scaling, moving, rotating) the objects concerned.

The topological neighborhood graph in Figure 4.2 allows us to define distance between any two topological relations, which can be used to measure the degree of topological relation similarity. According to the neighborhood graph, the length of the shortest path from TR_i to TR_j is defined as the distance $\mathbf{Dt(TR_i, TR_j)}$ between them where TR_i and TR_j are two topological relations. The topological distances among the eight topological relations are shown in Table 4.1.

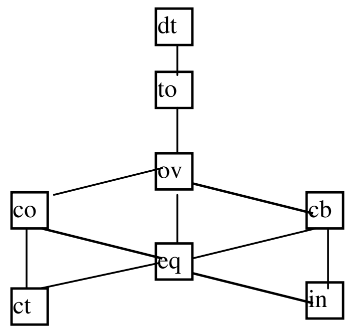


Figure 4.2. Topological neighborhood graph

Table 4.1 . Topological distances $Dt(TR_i, TR_j)$

TR_i \ TR_j	dt	to	ov	co	cb	eq	ct	in
dt	0	1	2	3	3	3	4	4
to	1	0	1	2	2	2	3	3
ov	2	1	0	1	1	1	2	2
co	3	2	1	0	2	1	1	2
cb	3	2	1	2	0	1	2	1
eq	3	2	1	1	1	0	1	1
ct	4	3	2	1	2	1	0	2
in	4	3	2	2	1	1	2	0

Topological information alone is insufficient to measure spatial relations because orientation is also a kind of important information in a spatial relation representation. In fact, the orientation domain is fuzzier than topological domain as we can see in the next section.

4.3.1.2 Structure of the orientation domain

The notion of neighborhood for the orientation domain is a little different from the topological domain due to the fuzziness in orientation relations. Two notions for the orientation neighborhood are used to describe the orientation domain as follows.

Definition 4.3. Two normal (fuzzy) orientation relations are *neighbors* if anyone of them can be transformed into the other by continuously deforming the concerned objects without going through another normal (fuzzy) orientation relation.

Definition 4.4. A normal orientation relation (a relation represented by a normal relation code such as rf, lf, and l etc) and a fuzzy orientation relation (a relation represented by a fuzzy relation code such as r|rf, rf|f, and f|lf etc) are *kin* if they are partially overlapped.

To differentiate neighbor and kin in the neighborhood graph, we use two different types of lines. A solid line in Figure 4.3 represents being a neighbor, and a dotted line represents being a kin. We use **Do(OR_i, OR_j)** to represent the orientation distance between two orientation relations OR_i and OR_j. If both OR_i and OR_j are the normal orientation relations or both are fuzzy orientation relations, **Do(OR_i, OR_j)** is defined as the length of the shortest path from OR_i to OR_j. However, if only one of OR_i and OR_j is a fuzzy orientation relation, then **Do(OR_i, OR_j)** is defined to have a value which is one less than the length of the shortest path between OR_i and OR_j. This is because we allow two kin nodes to be overlapped, and hence the distance between them should be defined as zero. For example, both r|rf and rf|f are overlapped with the orientation rf. Therefore Do(rf|f, rf)=0 and Do(r|rf, rf)=0. Similarly, we have Do(r|rf, r)=0. However, we should not infer from the previous two equalities that Do(r, rf)=0 because the overlapped orientation relation is not transitive. Do(r,rf) should be 1 according to our definitions.

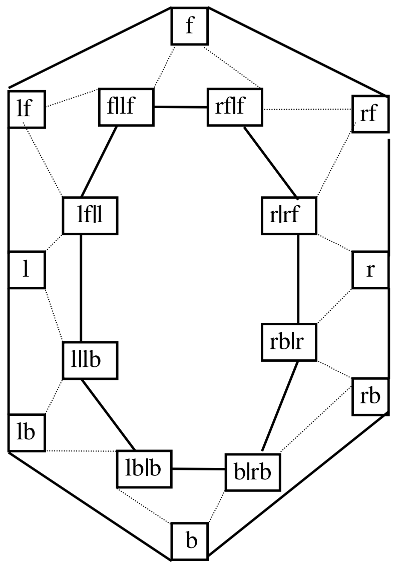


Figure 4.3. Orientation neighborhood graph

The complete set of orientation distances is shown in Table 4.2.

Table 4.2. Orientation distances Do(OR_i, OR_j)

<table>
<tr><th>OR_i \ OR_j</th><th>r</th><th>r|rf</th><th>rf</th><th>rf|f</th><th>f</th><th>f|lf</th><th>lf</th><th>lf|l</th><th>l</th><th>l|lb</th><th>lb</th><th>lb|b</th><th>b</th><th>b|rb</th><th>rb</th><th>rb|r</th></tr>
<tr><td>r</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>r|rf</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td></tr>
<tr><td>rf</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td></tr>
<tr><td>rf|f</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td></tr>
<tr><td>f</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td></tr>
<tr><td>f|lf</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td></tr>
<tr><td>lf</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>lf|l</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>l</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td></tr>
<tr><td>l|lb</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td></tr>
<tr><td>lb</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td></tr>
<tr><td>lb|b</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td></tr>
<tr><td>b</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>b|rb</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>rb</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>rb|r</td><td>0</td><td>1</td><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>4</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>

4.3.1.3 Degree similarity measure

In the degree similarity measure, we use the concept of neighborhood to measure the degree of changes in spatial relations. We are now ready to compute the total similarity distance **DTP(T1, T2)** (Distance for TuPles) between two IO&T tuples T1= $<O_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], O_j>$ and T2= $<O_x, [TR_{xy}, OR1_{xy}, OR2_{xy}], O_y>$, where O_i and O_j are the same objects as O_x and O_y respectively in two different pictures. It is defined as follow:

$$\mathbf{DTP(T1, T2) = Wt * Dt(TR_{ij}, TR_{xy}) + Wo * (Do(OR1_{ij}, OR1_{xy}) + Do(OR2_{ij}, OR2_{xy}))} \quad \mathbf{(4.2)}$$

where

Wt and Wo are used to denote the weights for topological distance and orientation distance respectively):

It should be noted that this is only one of the many ways that can be used to define DTP(T1, T2). It is quite reasonable to use a different weighted combination of the topological distance and the orientation distance according to the requirements of applications. However, in the following discussion, we use 1 for both Wt and Wo for the ease of discussion at this stage. The weight adjustment will be discussed later.

To measure the distance between two pictures, we only need to sum up all distances of the related spatial relation tuples, i.e. the similarity distance between a query picture Q and a database picture P is defined as:

$$\mathbf{DP(Q, P) = \sum_{j=1}^{m} DTP(T_{qj}, T_{pj})} \quad \mathbf{(4.3)}$$

Where there are m tuples for the query picture Q, T_{qj} is jth tuple in the query picture's IO&T representation, and T_{pj} is the corresponding tuple in the representation of the database picture P.

Definition 4.5. Degree similarity measure is a similarity measure based on similarity distance defined in (4.3). Similarity of the degree similarity measure is 1/DP(Q, P).

Let pictures P and Q be represented by (Vp, Rp) and (Vq, Rq) respectively where Vp and Vq are sets of object symbols, and Rp and Rq are sets of IO&T spatial relation tuples. The following algorithm can be used to calculate the distance DP(Q, P) between them.

Algorithm 4.4: Distance calculation function DP(Q, P)

Input: (Vq, Rq) and (Vp, Rp), the IO&T representations of the query picture Q and a database picture P.

Output: DP(Q, P), the distance between Q and P.

begin

1. *DP=0*
2. *if* $Vq \subseteq Vp$ *then*
3. *for each* $T1 = \langle O_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], O_j \rangle$ *in Rq*
4. *find* $T2 = \langle O_m, [TR_{mn}, OR1_{mn}, OR2_{mn}], O_n \rangle$ *in Rp*
   such that $O_m = O_i$, *and* $O_n = O_j$ *or* $O_m = O_j$, *and* $O_n = O_i$
5. *compute DTP(T1, T2)*
6. *add DTP(T1, T2) to DP*
7. *endfor*
8. *elseif set DP to* $\propto$
9. *endif*
10. *return DP*

end

Given two pictures p1 and P2, if DP(Q, P1) is smaller than DP(Q, P2), then P1 is more similar to Q. The complexity of this algorithm is $O(n^2)$ where n is the number of objects in the query picture.

In a real application, using either the number of matched tuples or degree of pair-wise spatial relation changes alone will produce ambiguous result. For example, in Figure 4.4, we have a query picture Q and two database pictures P1 and P2. We ignore the orientation relations and only consider topological relations for the ease of discussion. When P1 and P2 are compared with Q, 4 topological relations (those between A and each of B, C, D, and E) are changed in P2, but only 2 topological relations (those between B and C and between D and E) are changed in P1. So P1 is considered to be more similar to Q according to pair-wise spatial relation measure. However, it can be argued that P2 should be regarded as more similar to Q because there is only one object's position changed in P2. If we move A up a bit in P2, we will obtain Q. This anomaly occurs because the number of changes in pair-wise spatial relations multiplies even when only very few objects change their positions.

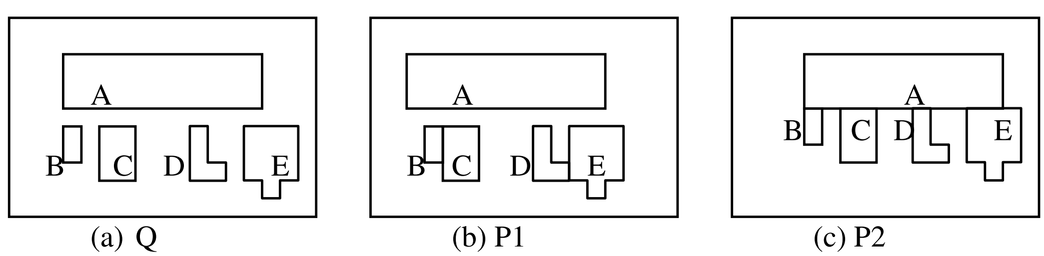


Figure 4.4. Query picture Q and two database pictures P1 and P2

This example shows that the question of "which similarity measure is more effective" does not have a simple answer because the definition of similarity is always application dependent. That is why so far there is no metric commonly adopted to measure the similarity. In the next section, we propose a simulated similarity retrieval that makes use of spatial relation changes for each object to simulate the common clique as an alternative similarity measure.

4.3.2 Simulated similarity retrieval

As we can find from the previous sections, pair-wise spatial relation measure and degree of spatial relation retrieval are practical because their algorithm complexity are $O(n^2)$. Therefore, it is not difficult for us to get values for these two measures. The simulated similarity retrieval in this section is to explore whether we may get the value of the common clique similarity retrieval heuristically because the complexity of the common clique algorithm is NP-hard.

Initially, we want to get the number of misplaced objects from the number of pair-wise spatial relations changes heuristically. This is possible when the number of spatial relation changes is mostly contributed by a small number of objects that have changed their spatial relations.

As the first step, those pictures that are very dissimilar should be weeded out with the application of any pair-wise spatial relations metrics discussed in the previous sections. After the filtering step, what remain will be subjected to the following test.

1. Calculate the number of spatial relations changes for each object in the query picture.
2. Sum all spatial relation changes based on each object and divide the sum by 2. Because the sum of all the numbers obtained in Step 1 is twice as large as the total number of spatial relation changes, the result is the total number of spatial relation changes.

3. Calculate the average number of spatial relation changes per picture and multiply it by a constant to get the tolerance T. Compare T with the number obtained for each object in Step 1. Count the number of objects whose number of spatial relation changes violates the tolerance.

Through the above three steps, we may get a rough estimate of the number of objects which cause many spatial relation changes. Tightening the tolerance will produce a more accurate estimate. The motivation in the use of such a tolerance in the above testing is as follow.

We assume the query picture Q has N objects and each database picture has at most n objects being misplaced, $n<<N$. For each of N - n static objects, there are at most n associated spatial relations changed. For each of the n displaced objects, there are at most N-1 associated spatial relations changed. Therefore, in general, an object with many spatial relation changes corresponds to one that has been displaced. With the use of an appropriate tolerance value, these objects can be identified.

Before we study the simulated similarity retrieval algorithm, we need to define the *similarity* of IO&T tuples and the related components.

Definition 4.6. The topological (orientation) relations between two spatial objects are *similar* if they have the same topological (orientation) code.

Definition 4.7. Two IO&T tuples, T1= $<O_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], O_j>$ and T2= $<O_x, [TR_{xy}, OR1_{xy}, OR2_{xy}], O_y>$, are *similar* if O_i and O_j are the same objects as O_x and O_y respectively, topological relations Tr_{ij} and Tr_{xy} are *similar*, and orientation relations $OR1_{ij}$ and $OR2_{ij}$ are *similar* to $OR1_{xy}$ and $OR2_{xy}$ respectively.

To compare two symbolic pictures with a small number of objects whose spatial relations have changed, we define the degree of similarity for the simulated similarity retrieval as follows.

Definition 4.8. The similarity distance DS(P, Q, T) between two pictures P and Q is the number of objects whose number of associated pair-wise spatial relations changes are bigger than a predefined tolerance T. Simulated similarity measure is a similarity measure using the similarity distance function DS(P, Q, T) such that the similarity of the simulated similarity measure is 1/DS(P, Q, T).

Thus a picture with smaller DS( ) value is regarded as one that is more similar to the query picture. Let pictures P and Q be represented by (Vp, Rp) and (Vq, Rq) respectively where Vp and Vq are sets of object symbols, and Rp and Rq are sets of IO&T spatial relation tuples. The following algorithm calculates the degree of similarity DS(P, Q, T) where P and Q are two pictures and T is a predefined tolerance.

Algorithm 4.5: Similarity function DS(Q, P, T)

Input: (Vq, Rq), the IO&T representation of the query picture Q;

(Vp, Rp), the IO&T representation of a database picture P;

N, the number of objects in Q;

T, a predefined tolerance

Output: DS(Q,P,T), the similarity distance between Q and P based on the threshold T

begin

1. *DS=0;*
2. *for each object O_i in Vq set TMP[O_i]=0;*
3. *if Vq$\subseteq$Vp then*
4. *for each T1=<O_i, [TR_{ij}, $OR1_{ij}$, $OR2_{ij}$], O_j> in Rq*
5. *find T2=<O_m,[TR_{mn}, $OR1_{mn}$, $OR2_{mn}$],O_n> in Rp*

such that $O_m=O_i$, and $O_n=O_j$;

6. *if T1 is not similar to T2*
7. *$TMP[O_i]=TMP[O_i]+1$; $TMP[O_j]=TMP[O_j]+1$;*
8. *endif*
9. *endfor*
10. *elseif set DS to $\propto$*
11. *for each object O_i in Vq*
12. *if ($TMP[O_i]>T$) then $DS=DS+1$;*
13. *return DS*

end

Usually T can be set according to the total number of spatial relations for each object in the query picture and it is application dependent. The complexity of the algorithm is $O(n^2)$ where n is the number of objects in Q. Compared with the common clique algorithm, this simulated similarity algorithm is certainly more practical to get the number of misplaced objects. It is very likely that the set of static objects form a clique, and those misplaced objects are in the complement set of the clique.

4.4 Object based similarity retrieval

In previous sections, we have discussed several similarity retrieval measures. These functions measure certain aspect of similarity as follows:

- How many objects have changed their spatial relations (Some spatial relations have been changed due to the movements of these objects);
- How many pair-wise spatial relations have changed (The number of changed pair-wise spatial relations);
- To what degree the pair-wise spatial relations have changed (The extent of the changes based on a defined measure).

As we mentioned early, the question of which similarity measure is more effective is not easy to answer because the definition of similarity is always application dependent. The relations of the real answer set with the answer sets derived by the above three measures are illustrated in Figure 4.5. As we can see, although three answer sets found by the three measures overlap with the real answer set, none of them is exact. However, if we could not get the real answer set exactly, we may want to settle with an answer set which can cover the real answer set. In this section, we propose to use a reference object in the measurement of similarity to find this kind of covering answer set. The new measure is developed from the object based similarity concept.

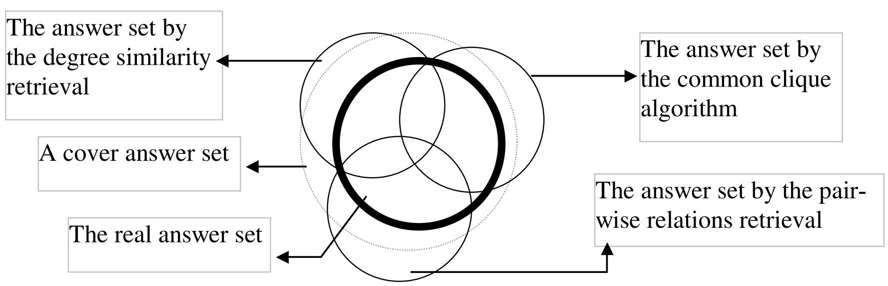


Figure 4.5. Relations between the real answer and those derived from the use of different measures

4.4.1 Object based similarity retrieval

In real life, a query about a scene is always issued with reference to some *dominant objects*. For example, in the query “to find a house, in the front of that there is a temple and a swimming pool both facing this house”, the house is the dominant object. The spatial relation between the temple and the swimming pool is not mentioned and is assumed to be not important. To satisfy the query, we may find all database pictures based on the spatial relations of a house with the other two objects. When there are several dominant objects, we may base on the first dominant object O1 to retrieve the set S1 of similar pictures within a tolerance, and then the second dominant object O2 to retrieve similar pictures from S1 to form S2. This process is repeated until the last dominant object has been considered. The last set obtained is the answer set of pictures that are most similar to the query picture based on all the dominant objects that have been arranged in an order according to how dominant they are. This is *object based similarity retrieval*. The retrieval is based on dominant objects. A query picture in which all objects are dominant is only a special case of the object based similarity retrieval.

Incorporating the concept of dominant objects, we redefine the notation of similarity based on the IO&T representation.

Definition 4.9. The IO&T similarity measure DP is a vector function that is defined for a database picture P, a query picture Q and a dominant object O, such that the similarity distance based on objects in the picture **DOP(O, Q, P) = (D1(O, Q, P), D2(O, Q, P))** where D1(O, Q, P) is the number of objects that have changed their spatial relations with respect to O and D2(O, Q, P) measures the total extent of the changes.

Definition 4.10. D1(O, Q, P) is the number of objects (other than O in the query picture) whose corresponding spatial relation tuple involving object O is changed in the database picture P.

D2(O, Q, P) is closely related to the function DTP( ) which is used to measure the distance between two IO&T tuples in the previous sections.

$$\mathbf{D2(O, Q, P) = \sum_{j=1}^{n-1} DTP(T_{qoj}, T_{poj})} \qquad \mathbf{(4.4)}$$

Where n is the number of objects in the query picture Q, T_{qoj} is a tuple including object O in the representation of the query picture Q. T_{poj} is the respective tuple in the IO&T of the database picture P. There are n-1 of such kind of tuples for a query picture.

Let pictures P and Q be represented by (Vp, Rp) and (Vq, Rq) respectively where Vp and Vq are sets of object symbols, and Rp and Rq are sets of IO&T spatial relation tuples. The following algorithm calculates the IO&T similarity measure.

Algorithm 4.6: Function for similarity measure DOP(O, Q, P)

Input: O, a dominant object;

(Vq, Rq) and (Vp, Rp), the IO&T representations of the query picture Q and a database picture P.

Output: D1(O, Q, P), the number of objects that have changed their spatial relations with respect to the dominant object O in Q and P;

D2(O, Q, P), the total extent of changes with respective to the dominant object O.

begin

1. *D1=0, D2=0*
2. *if* $Vq \subseteq Vp$ *then*
3. *for each* $T1=<V_i, [TR_{ij}, OR1_{ij}, OR2_{ij}], V_j>$ *in Rq*
4. *if* V_i *or* $V_j = O$ *then begin*

5 *find* $T2=<U_m, [TR_{mn}, OR1_{mn}, OR2_{mn}], U_n>$ *in Rp such that* $U_m=V_i$*, and* $U_n=V_j$

6. *compute DTP(T1, T2)*
7. *if DTP(T1, T2)<>0 then add 1 to D1 and add DTP(T1, T2) to D2*
8. *endif*
9. *elseif set D1 and D2 to* $\propto$
10. *return D1 and D2*

end

D1 and D2 are used to determine whether P is similar to Q. If D1=0 and D2=0, then a sub-picture of the database picture P matches exactly with the query picture Q based on the dominant object O, and hence P should be accepted. Usually the acceptance criteria will be relaxed by setting some tolerance for D1 and D2 according to the requirements of an application. When D1 and D2 are both smaller than their respective tolerance, then P will be accepted. When there are k dominant objects, the same algorithm will be invoked k times with each invocation working on the output list of similar pictures from the previous invocation.

If we relax the tolerances for D1 and D2, then there are always more pictures being retrieved. We order the similar pictures based on the values of a weighted combination of D1 and D2. The precise definition of such a ranking function will be application dependent. The complexity of this algorithm is $O(n^2)$ where n is the number of objects in Q.

4.4.2 An example for object based similarity retrieval

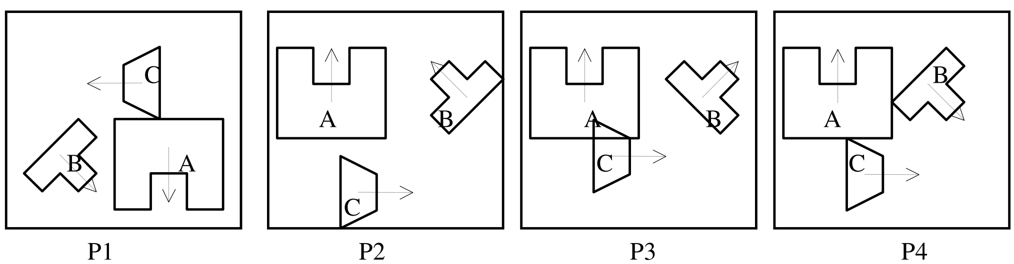


Figure 4.6. Example pictures (All arrows represent intrinsic fronts of objects)

We use the example in Figure 4.6 to illustrate the retrieval procedure. In Figure 4.6, P1 is a query picture whereas P2, P3, and P4 are three database pictures. First, we assume A to be the only dominant object in the query. After removing all the IO&T tuples not related to A, P1, P2, P3, and P4 will contain the following tuples:

P1: <A, [dt, r, lf|l], B>, <A, [to, b|rb, l|lb], C>

P2: <A, [dt, r, lf|l], B>, <A, [dt, b|rb, l|lb], C>

P3: <A, [dt, r, lb|b], B>, <A, [ov, b|rb, l|lb], C>

P4: <A, [to, r|rf, rb|r], B>, <A, [to, b|rb, l|lb], C>

Using Algorithm 4.6, the distances between P1 and other pictures can be computed as follows:

D1(A, P1, P2)=1, D2(A, P1, P2)=1;

D1(A, P1, P3)=2, D2(A, P1, P3)=3;

D1(A, P1, P4)=1, D2(A, P1, P4)=5;

Because D1(A, P1, P2) and D1(A, P1, P4) are both 1, we need to compare their respective D2. As D2(A, P1, P2) is smaller than D2(A, P1, P4), we conclude that P2 is the most similar picture to P1. It is obvious that P3 is the least similar picture to P1 because D1(A, P1, P3) is the biggest among the three D1s.

When both A and B are dominant objects, then all the IO&T tuples are involved. If the tolerance for D1 is 2 and the tolerance for D2 is ∝, then we are only concerned about the number of spatial relations changed with respect to the dominant objects. Using A as the first dominant object in the similarity retrieval, we get S1={P2, P4}. P3 is rejected because D1(A, P1, P3) is not smaller than the tolerance. The next similarity retrieval is from S1 using B as the dominant object. After removing all the IO&T tuples not related to B, P1, P2, and P4 are shown as follows:

P1: <A, [dt, r, lf|l], B>, <B, [dt, lf|l, lf], C>

P2: <A, [dt, r, lf|l], B>, <B, [dt, lf|l, f|lf], C>

P4: <A, [to, r|rf, rb|r], B>, <B, [dt, r|rf, lf], C>

Using Algorithm 4.6 again, the distances based on B can be computed as follows:

D1(B, P1, P2)=0, D2(B, P1, P2)=0;

D1(B, P1, P4)=2, D2(B, P1, P4)=8;

Since D1(B, P1, P4) is not smaller than the tolerance, P4 will be discarded. As a result, P2 is accepted as the only picture that is similar to P1 using A and B as the dominant objects. If we rotate P1 by 180 degrees, we find that P2 is the most similar picture to P1 among the three pictures (P2, P3, and P4) since only object C has moved down a bit in P2 and there is no change in other objects. The fact that we are able to retrieve the correct picture using the object-based similarity measure indicates that this approach of similarity retrieval is promising.

Up to now, we have discussed some similarity measures that include some intuitive measures, some derived measures and an integrated measure. Our main concern is whether retrieval algorithms based on these measures are accurate. Therefore, in the following, we implemented some experiments to compare the proposed similarity measures to understand the differences among them.

4.5 Empirical comparison of the proposed similarity measures

In this section, we will compare the proposed similarity measures based on the same representation i.e. IO&T. The additional comparison with other representations and respective measures will be continued in the next chapter.

As we discussed in previous sections, the common clique measure and the spatial relation measure are more of theoretical interest. They are not really the focus of our comparison though they will still be referenced in the following. There are three experiments implemented. The first one is to test if the simulated measure is feasible and what is a reasonable threshold for the metric. The second one is to compare the difference of degree similarity measure, simulated similarity measure, and spatial relation measure. Object-based similarity retrieval integrates both the number of object changes and the degree of changes. Therefore, the last experiment is to test if the object-based similarity measure is effective. At the end of the experiments, we will conclude what are the typical similarity measures used for our following discussion.

4.5.1 Feasibility testing of the simulated similarity measure

Since the simulated measure is derived from the common clique approach, we will test how accurate it is to simulate common clique measure using the simulated similarity measure. The detail setup of the experiment is as follow:

The experimental database will be generated automatically. The process of generating the database is based on query pictures. We will randomly move/rotate some objects in the query picture to generate a database picture. Since we can control the number of object changes, we can group the generated pictures into different groups based on the changes. Each experimental database consists of 600-700 pictures with each image contains 10 to 15 objects with extent. The pictures are divided into 5 groups according to the number of objects' changes (such as moving and rotating) with respect to the query picture. There are 20, 45, 120, 210, and 252 pictures in the group G1, G2, G3, G4, and G5 respectively for each database. A picture is in Gi when there are i objects changing their positions with respect to the query image. Pictures with five or more objects changing their positions belong to G5.

In Gi, since there are only i object(s) changing its location and i << n (where n is the number of objects in a picture), we should have the similarity distance (to the query picture) equals to i for any picture in Group i using the common clique measure (this is not really true since some movement of objects may not affect their spatial relation). For the simulated similarity measure, we then set the threshold T to different values such as 30%, 35%, 40% etc. Suppose we want to find all pictures from the database with similarity distance i, we will test how many pictures are from Group i, and how many of them are not from Group i in this experiment. To gauge the effectiveness of the retrieval algorithm, we define the recall rate and precision as follows:

Definition 4.11. Let g be number of all pictures in Group G, r be the number of pictures retrieved from G, and q be the number of all retrieved pictures (i.e. pictures retrieved from G and from other groups altogether). Then the recall rate R and the precision P are defined as

$$\mathbf{R = 100*r/g \quad and \quad P = 100*r/q} \tag{4.5}$$

The empirical results are shown in Table 4.3.

Table 4.3. Simulated similarity measure experiment results

Group	No of objects with changes	T=30%		T=35%		T=40%	
		R	P	R	P	R	P
1	1	100	100	95.6	88.7	96.4	70.8
2	2	100	100	93.3	84.0	88.9	63.5
3	3	100	99.2	93.3	88.2	81.7	58.7
4	4	99.5	100	92.9	91.5	70.0	62.0
5	5+	85.1	95.3	80.5	90.1	71.2	60.6

From Table 4.3, the simulated result shows that the simulated similarity measure is effective in retrieval for a small number of object changes, in particular, when the number of object changes is less than 5. This means the simulated measure is able to measure small changes for IO&T spatial relation representation of picture. On the other hand, we also find that the recall rates drop when more objects are swapped. This is because when the number of swapped objects increases, our assumption that this number should be much smaller than the number of objects in the query picture will no more be valid. We also observe that T's setting will affect both the recall rate and the precision. Therefore, the selection of T is very important. In our experiment, the 30% of the total number of spatial relations for each object is most satisfactory. However, this value may be changed for different applications.

4.5.2 Comparison of derived similarity measures

The simulated similarity measure looks good in the previous experiment because the experiment itself is in favor of the number of object changing. However, it is possible that the changes of very few objects can cause drastic changes in our perception (such as the example in Figure 4.4). The simulated similarity measure has a restriction that the number of changed objects must be much smaller than the total number of objects in a picture. On the other hand, the simulated similarity measure needs some training process to gauge the appropriate threshold value needed. Therefore, in this section, we will discuss another experiment to compare simulated similarity measure with spatial relation measure and degree similarity measure. The objective is to find out other factors affecting spatial similarity and advantages of different similarity measures.

Based on the previous test, we choose 30% as the threshold for the simulated similarity measure. The idea is to retrieve top 10, 20, 30, and 40 similar pictures from each pictorial database, and check the picture distribution among all groups to see the difference for different measures in Table 4.4, Table 4.5, and Table 4.6 respectively.

Table 4.4. Distribution of picture retrieved using degree similarity measure

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	100.00	0.00	0.00	0.00	0.00
20	75.00	25.00	0.00	0.00	0.00
30	60.00	23.33	16.67	0.00	0.00
40	50.00	25.00	20.00	5.00	0.00

Table 4.5. Distribution of picture retrieved using spatial relation measure

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	90.00	10.00	0.00	0.00	0.00
20	80.00	20.00	0.00	0.00	0.00
30	63.33	30.00	6.67	0.00	0.00
40	50.00	32.50	15.00	2.50	0.00

Table 4.6. Distribution of picture retrieved using simulated similarity measure

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	100.00	0.00	0.00	0.00	0.00
20	100.00	0.00	0.00	0.00	0.00
30	63.33	26.67	0.00	0.00	0.00
40	50.00	45.00	2.00	0.00	0.00

From Table 4.6, we may notice that the top 10, 20 similar pictures are all from Group 1, and the top 30 and top 40 similar pictures are mostly from Group 1 and Group 2. This means that the return sets of the approach using the simulated similarity measure are more restricted from the first few groups, i.e. those groups with less number of objects involving movements. This can be accepted in the context of a very small number of movements affecting less spatial relations, in particular for pictures in Group 1. However, there should be cases that a big portion of spatial relations has been changed due to some drastic changes of a few objects in our generated iconic picture database, which are not detected effectively by the simulated similarity measure.

Relatively, from Table 4.4 and Table 4.5, the distribution of returned similar pictures is boarder for both degree based measure and spatial relation measure, in particular, there are more returned pictures from Group 3, or even Group 4 in the top 30 and top 40 similar pictures. This means that these two similarity measures can detect spatial relation changes instead of just the number of objects involving changes. However, compare to spatial relation measure, degree based measure is also reasonable good for detecting less number of objects involving movement whilst spatial relation measure is not. This can be observed from the results in Table 4.4 and Table 4.5. The approach using the degree measure will detect all pictures from Group 1 (i.e. the group with only one object involving movement) compare to the 90 percent of the approach using the spatial relation measure. In fact, this proves that degree based measure is more compatible and flexible comparing to the other two similarity measures. Degree based measure provides finer granularity in its measurement which is good for measuring approximate similarity. Therefore, in the following discussion, degree based measure will be always used as one of major measures for comparison of similarity retrieval approaches.

4.5.3 Feasibility of object-based similarity measure

In previous experiments, we have noticed that both the number of object involving movement and the extent of the spatial relation changes can be useful in the measurement of the spatial relation for iconic pictures. However, in real life, we only consider objects that are important. Therefore, we introduced the object-based similarity retrieval (which bases on dominant objects) on top of the proposed retrieval measures. In this section, we continue our experiment with object-based approach to see how effective it is to use the object-based similarity concept with proposed measures.

We still use the same experiment database used in the previous section. Since there are at most 15 distinct objects in the query picture, the number of spatial relation tuples involving a dominant object is 14. We set the tolerance Tol1 for D1 (the number of objects that have changed their spatial relation with respect to the dominant object) to 15, 20, and 25 percent of 14 tuples that are 2, 3, and 4 tuples respectively. Referring to the definition of DTP( ) which measures the distance between IO&T tuples, a spatial relation tuple can attain the maximal extent of changes in its topological relation and its orientation relations with a value of 12 if both weight constants Wt and Wo in DTP are set to 1. When all 14 tuples involving the dominant object attain the maximum, the total changes will be $14 \times 12 = 168$. The tolerance Tol2 for D2 (the total extent of changes that have taken place) is set to 3, 6, 9, 12 and 15 percent of 168 which are 5, 10, 15, 20 and 25 respectively. To gauge the effectiveness of the retrieval algorithm, we randomly choose one of the 5 unchanged objects as our dominant object of retrieval, and calculate for each group the average percentage of pictures that have been retrieved. The results are shown in Table 4.7.

Table 4.7. Object-based similarity retrieval experiment results

Tol1	Tol2	The percentage of pictures retrieved (%)			
		Group 2	Group 3	Group 4	Group 5
5%(2)	3%(5)	45.78	20.00	7.05	1.83
	6%(10)	67.56	30.67	11.43	3.57
	9%(15)	87.11	37.83	13.71	4.13
	12%(20)	99.56	42.00	14.76	4.29
	15%(25)	100.00	42.00	14.76	4.29
10%(3)	3%(5)	45.78	20.00	7.05	1.83
	6%(10)	67.56	44.00	23.33	11.19
	9%(15)	87.11	65.67	38.00	19.76
	12%(20)	99.56	96.67	52.95	24.60
	15%(25)	100.00	100.00	54.10	24.84
15%(4)	3%(5)	45.78	20.00	7.05	1.83
	6%(10)	67.56	44.00	25.24	13.10
	9%(15)	87.11	65.67	49.33	33.10
	12%(20)	99.56	96.67	88.38	59.84
	15%(25)	100.00	100.00	98.67	65.71

From Table 4.7, we find that the percentage of pictures retrieved from groups with fewer object changes is always higher. When we relax both the tolerances Tol1 and Tol2 for D1 and D2 respectively, there are more and more pictures being retrieved. Since each picture in Group i (i=2 to 5) has at most i tuples changed with respect to each dominant object, by setting Tol1≥i, the percentage of pictures retrieved from Group i will be determined by the magnitude of Tol2 alone. From Table 4.7, we observe that by setting Tol1 to i+1 and Tol2 to 25(15%), we can always retrieve more than 98% pictures from Group i. However, there are quite many pictures being retrieved from other groups as well. For example, when Tol1=3 and Tol2=25, there are about 25% of the pictures from Group 5 found to be similar. This shows that even when many objects have shifted their positions, as long as their extent of changes remain small, they will be retrieved as pictures similar to the query pictures, yet they may be rejected when the common clique measure or the simulated similarity measure is being used.

This experiment shows clearly again that the number of objects that have changed their positions is not the only factor which affects the spatial similarity. Although in real life, a picture with many objects changed their positions will be more likely to be rejected in similarity retrieval, it should be accepted if the extent of changes is small. Compare to the previous experiments without considering dominant objects, we can also control the similarity retrieval returns using the two thresholds (i.e. the threshold for the number of object change, and the threshold for the total extent changes in spatial relations). Therefore, the use of object-based similarity measure makes it possible to retrieve similar pictures with due consideration for both factors, which is more flexible and adaptive.

4.6 Summary

In this chapter, we have discussed similarity metrics that are used to measure the spatial similarity based on the representation IO&T proposed in the previous chapter. Native similarity measures, i.e. common clique measure and spatial relation measure, are discussed as the base concept for more practical measures. Simulated similarity measure and degree similarity measure are proposed to cater for different retrieval objectives in different applications. It is highlighted that none of the discussed approaches is complete or perfect for every need from the users. Therefore, we introduced the idea of object-based similarity retrieval to overcome the fuzzy part of the similarity concept and integrate the advantages of previous measures. In general, the simulated similarity measure is good for similarity retrieval involving a small number of object involving movements whilst the degree similarity measure is more sensitive to detect the extent of spatial relation changes. Object-based similarity retrieval approach on top of the proposed measures can effectively integrate the advantages of both approaches by selecting the dominant objects in an intuitive way. However, all the similarity measure discussions so far are based on the proposed IO&T representation. In the following chapter, we will study some components in spatial relation representation and similarity measure that have been missing in the discussion. In particular, the extension of an object is purposely ignored so far. The extension of an object refers to the distance among iconic objects and the size of iconic objects. Since the extension of an object is hardly interpreted by a pure qualitative representation, an augmented spatial relation representation is proposed to retain the invariant property of the current representation and to include the extension of an object and the distance between objects in a fuzzy way. We will also discuss some experiments comparing several different representations and their respective similarity measures for automatic similarity retrieval in the next chapter.

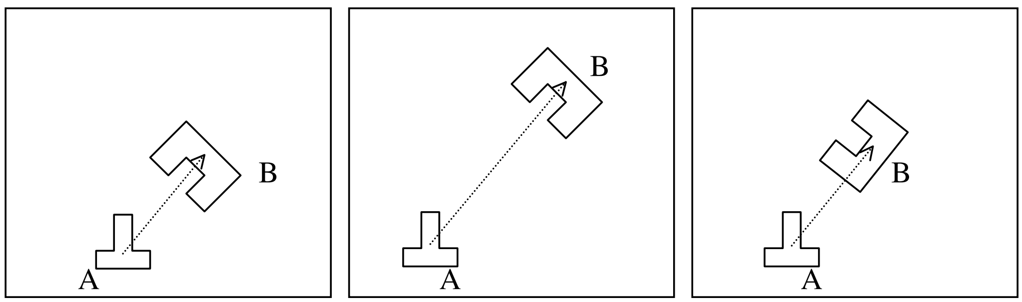
Chapter 5

Beyond qualitative spatial relation representation and retrieval

In previous chapters, we have discussed spatial relation representations, and similarity measures to compare the similarity between pictures. In this chapter, we will continue to discuss an augmented spatial relation representation and retrieval. We will also compare the similarity retrieval solutions based on different representations and respective measures at the end of this chapter.

5.1 Augmented relation representation and retrieval

Up to now, we have introduced a complete solution for automatic spatial relation similarity retrieval process. However, some information is not captured in our proposed representation, i.e. the extension of objects. Usually orientation categorization is based on the relative positions of the centroids of objects. However, for objects with extension, this kind of orientation relation is inadequate. For example, in Figure 5.1(a), if we simply say object B is at the northeast of object A according the centroids of A and B, then there will be no difference between Figure 5.1 (b), (c) and (a). However, from a human's viewpoint, they are different. In Figure 5.1(b), the relative distance between object A and B is different from that in Figure 5.1(a). The intrinsic orientations of object B in Figure 5.1(a) and Figure 5.1(c) respectively are different as well. To capture the missing information about an object's extent, we introduce the **Augmented Orientation Spatial Relation (AOSR)** representation.



(a) (b) (c)

Figure 5.1. Example pictures (a), (b), and (c)

As the name suggests, AOSR is to capture additional information, which is the extent of object essentially. However, the extent of object is typically quantitative information that tends to be inflexible. If we use qualitative approach to represent the extent of object (such as "big", "small", "far", and "near" etc), the representation will be highly subjective, and is not really practical in real applications. On the other hand, as we mentioned in the previous chapter, the objective of our representation for similarity retrieval should have the following properties:

- Accurate and flexible;
- Invariant under transformation (such as rotation, enlargement etc);
- With a feasible similarity measure for approximate retrieval.

Therefore, AOSR is actually not a purely qualitative representation when the extension of object is included.

5.1.1 Definition of AOSR

The augmented orientation spatial relation representation (AOSR) between a reference object and a primary object is defined as follows:

Definition 5.1: The **augmented orientation spatial relation (AOSR)** between a reference object O_r and a primary object O_p is a tuple of α and β, where α and β are the two angles measured from the begin-bound and the end-bound respectively. The begin-bound is the first tangent line which is about to enter object O_p when rotating anti-clockwise a half line originated from the centriod of object O_r. The end-bound is the first tangent line which is about to leave object O_p when rotating anti-clockwise a half line originated from the centriod of object O_r. The starting direction of the rotating half line is the **intrinsic front** of object O_r. The spatial relation between O_r and itself is defined to be NULL. β must be bigger than α.

The illustration of the above definition is shown in Figure 5.2. With this definition, we may capture the orientation spatial relations of objects with extension more precisely because the begin-bound angle and the end-bound angle denote the extension of the primary object in the polar coordinate system of the reference object. In the meantime, the relative distance information of the primary object and the reference object can also be reflected by the begin-bound angle and the end-bound angle combined with the intrinsic orientations of both objects. In Figure 5.3, when we move object O_p away or toward object O_r along the line between the two centriods of O_r and O_p, the begin-bound angle and the end-bound angle will change. The relative distance of O_r and O_p can be reflected by the difference β - α if both O_r and O_p's intrinsic fronts remain unchanged.

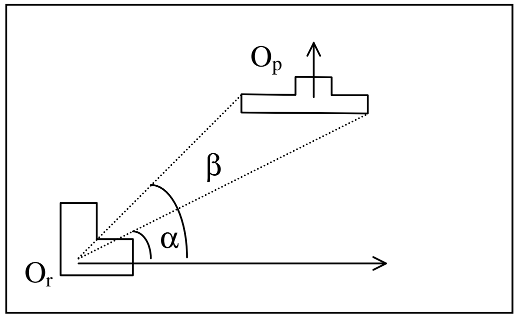


Figure 5.2. Definition of α and β for O_r and O_p (The arrows represent the intrinsic fronts of objects)

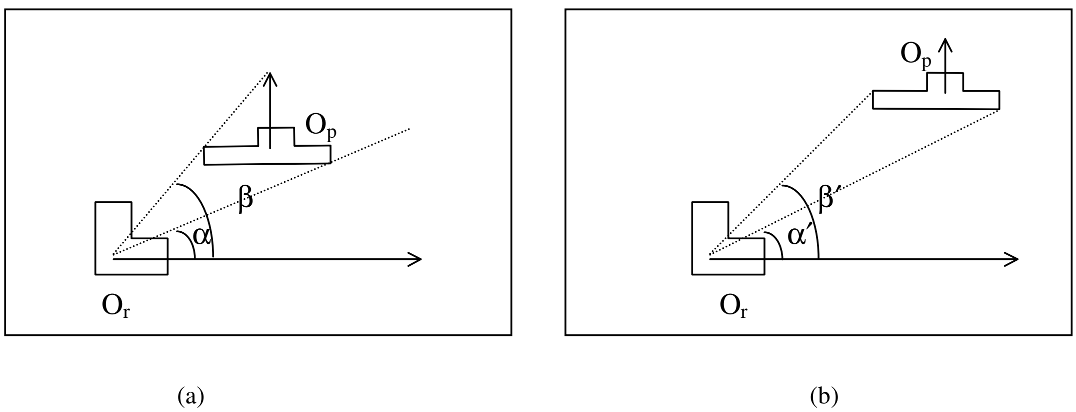


Figure 5.3. An example (the arrow represents intrinsic front of objects)

However, usually the begin-bound angle and the end-bound angle will change too when the orientation of object O_p is changed. An example is shown in Figure 5.4. This will cause ambiguity because either the change in relative distance and orientation may result in the same AOSR for a reference object O_r and a primary object O_p. The problem can be solved if we use **intrinsic orientation**. For example, with the use of **intrinsic orientation** of the reference object as the reference frame, even if the begin-bound angle and the end-bound angle are the same, the change in the intrinsic orientation of the primary object O_p in Figure 5.4 can still be detected. This will cause the orientation spatial relation to change when O_p is used as reference object. Hence the intrinsic orientation used in the AOSR plays an important role in measuring the angles.

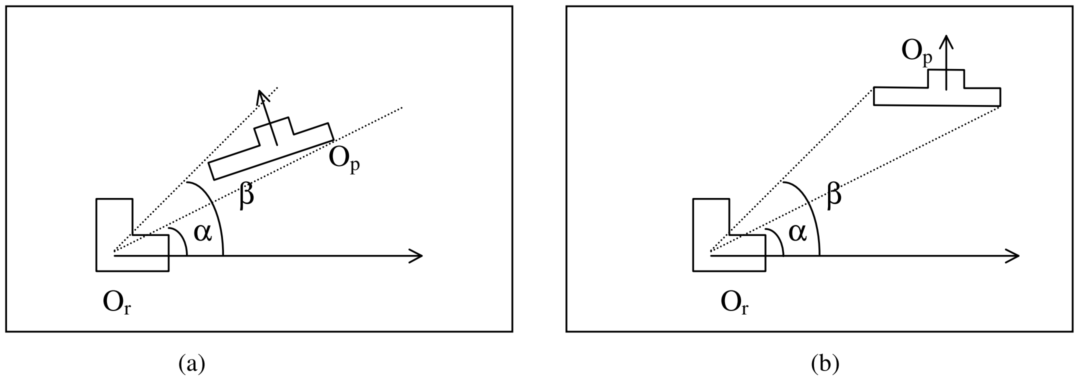


Figure 5.4. An example (the arrow represents intrinsic front of objects)

In the above, what we have defined is the AOSR between a primary object and a reference object. Given a picture with n different objects, each of the n objects can be chosen as a reference object with the rest as primary objects. Therefore, the AOSR of a picture is defined as follows:

Definition 5.2: For a picture with n different objects, its **augmented orientation spatial relation (AOSR)** representation is an n × n matrix where each row is the AOSR between the row index object as the reference object and each of the n objects as the primary objects. The row index objects and the column index objects are lexicographically sorted.

As an example, the AOSR of picture P in Figure 5.5 is as follow (For the ease of illustration, all angles are of integer values):

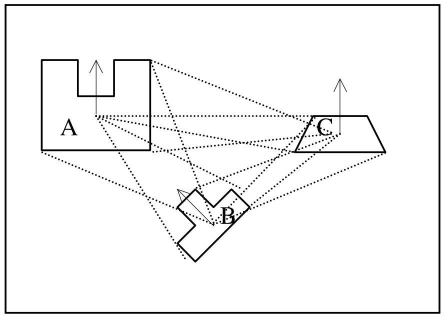


Figure 5.5. An example picture P (all arrows represent intrinsic fronts of objects)

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c|lll|} & A & B & C \\ \hline & NULL & (210, 240) & (255, 270) \\ & (-15, 15) & NULL & (240, 270) \\ & (45, 95) & (210, 235) & NULL \end{array}$$

Note that P(B, A) = (-15, 15) by following the rule that β must be bigger than α as stated in definition 5.2.

5.1.2 Handling non-regular objects

In some cases, there is a mixture of regular and non-regular objects (without intrinsic front, such as a ball). To construct the AOSR matrix, we need to handle the following two cases:

1. One of the objects concerned is non-regular.

   In this case, we use the regular object's intrinsic front as the non-regular object's intrinsic front.

2. Both objects are non-regular objects.

   In this case, α is always 0 and β is the subtended angle of the primary object with respect to the centroid of reference object.

It is very straightforward to use only existing regular objects' intrinsic orientation as the reference frame in the AOSR matrix since non-regular object's self-rotation will not affect the spatial relation between objects usually. However, for pictures containing only non-regular objects, our AOSR is not applicable. Therefore, we assume all pictures contain at least one regular object. A special case is that there is only one regular object in a picture. In this case, the intrinsic orientation of the regular object is the only reference orientation frame in the picture.

5.2 Similarity retrieval for AOSR

Since similarity is fuzzy, it may be necessary to use different similarity functions for different comparison objectives. The AOSR is defined based on object, and hence the retrieval procedure discussed will also be based on object. In fact, this follows human retrieval procedure in real life. To compare two pictures, we normally compare them by portions using different object for reference in each comparison.

Without loss of generality, consider two symbolic pictures Q and P with Q the query image and P the database image. To define a similarity based on object O_1 in Q, the first step is to check whether P contains O_1. The second step, the corresponding rows and columns indexed by O_1 in the matrices of P and Q are used to compute the similarity. Following Gudivada's [Gudi95] assumption, the maximum similarity is set to 100. If there are n objects in Q, each of the n-1 tuples involving other objects' orientation with respect to O_1 contributes a value of 100/(n-1) toward the similarity based on object O_1. Suppose O_2 is another object that appears in both P and Q. $P(O_1, O_2)$ denotes the item indexed by O_1 and O_2 in picture P's AOSR matrix. If we have

$P(O_1, O_2) = (\alpha^P{}_{12}, \beta^P{}_{12})$,

$P(O_2, O_1) = (\alpha^P_{21}, \beta^P_{21})$,

$Q(O_1, O_2) = (\alpha^Q_{12}, \beta^Q_{12})$,

$Q(O_2, O_1) = (\alpha^Q_{21}, \beta^Q_{21})$,

then the four pairs of angles α^P_{12} and α^Q_{12} , β^P_{12} and β^Q_{12} , α^P_{21} and α^Q_{21} , and β^P_{21} and β^Q_{21} will be used to measure the similarity based on O_1 and O_2. When the angular difference between a pair of angles increases from 0 to 180, the similarity contributed by this pair of angles will decrease from the maximum to the minimum. However, when the difference between one pair of angle increases from 180 to 360, the similarity contributed by this pair of angles will increase from the minimum to the maximum. Hence, we choose the cosine value of the difference for the pair of angle plus 1 to ensure the value to be nonnegative. The contribution factor from O_2 to the similarity based on O_1 is defined as

$$\begin{aligned}\mathbf{S_{12} = 100*((\cos(\alpha^P_{12} - \alpha^Q_{12})+1) + (\cos(\beta^P_{12} - \beta^Q_{12}) + 1)}\\ \mathbf{+ (\cos(\alpha^P_{21} - \alpha^Q_{21}) + 1) + (\cos(\beta^P_{21} - \beta^Q_{21})+1) + 1)/9(n-1)} \qquad \mathbf{(5.1)}\end{aligned}$$

In formula (5.1), when $\alpha^P_{12} = \alpha^Q_{12}$, $\beta^P_{12} = \beta^Q_{12}$, $\alpha^P_{21} = \alpha^Q_{21}$, $\beta^P_{21} = \beta^Q_{21}$, the contribution factor is 100/(n-1), where n is the number of objects in the query picture. This is the maximum similarity contributed by one object. This is also the reason why there is a 9 in the formula as $((\cos(\alpha^P_{12} - \alpha^Q_{12})+1) + (\cos(\beta^P_{12} - \beta^Q_{12}) + 1) + (\cos(\alpha^P_{21} - \alpha^Q_{21}) + 1) + (\cos(\beta^P_{21} - \beta^Q_{21})+1) + 1)/9$ can be = 1 in this situation. When the difference between the corresponding angles in P and Q is bigger, the similarity contribution is smaller. It should be noted that the minimum similarity contribution is not 0 but 100/9(n-1) when all corresponding angles are different by 180 degree. However, if either O_1 or O_2 does not appear in P, the contribution factor from O_2 is defined as 0. This means that as long as O_2 appears in P, the contribution factor from O_2 to S_{12} must be bigger than 0, a very important characteristic of the similarity measure.

Now we may replace O_2 by any other object O_j appearing in Q to obtain S_{1j}. Then S_1, the total similarity based on object O_1, is the sum of all contributions from all other n-1 objects in Q.

$$\mathbf{S_1 = \sum_{j\neq 1} S_{1j}} \quad \textbf{(j= 1…n, where n is the number of objects in Q)} \qquad \textbf{(5.2)}$$

The maximum S_1 is 100 and the minimum S_1 is 0 when O_1 is missing from P or all other n-1 objects are missing. The value of S_1 indicates how spatially similar is P to Q based on O_1.

Suppose the AOSR matrices for Q and P are M_Q and M_P respectively. We define a similarity function SV:

$$\mathbf{SV(M_Q\,,\,M_P) = (S_1\,,S_2\,,\ldots\ldots,\,S_n)} \textbf{ where n is the number of objects in Q} \qquad \textbf{(5.3)}$$

The SV() uses M_Q and M_P as inputs to compute the similarity degrees between Q and P based on each of the objects in M_Q.

How to compute the spatial similarity between pictures by using the n similarity values produced by SV() is application dependent. Here are two possible approaches:

1. Select some **dominant objects** in the query image to compare similarity. For example, in the query to *find a house, in the front of that there is a temple and a swimming pool both facing this house,* the house is the dominant object. The

spatial relation between the temple and the swimming pool is not mentioned and is assumed to be unimportant. In this case, not all values from the SV() are useful. In a real application, it may not be necessary to compute all S_1 to S_n. This approach can trim the answer set of similar pictures very effectively. It is especially useful for similarity retrievals involving only a small number of dominant objects

2. Comparing the similarity based on the sum of S_1 to S_n. This approach is widely used for the image retrieval without dominant objects. The result is a small set of candidate images ranked according to their similarity degrees.

In the following, we have carried out some experiments on 2D-PIR, IO&T, and AOSR representations and measures for comparison.

5.3 Experiments on 2D-PIR, IO&T, and AOSR representations and measures

As discussed in Chapter 2, there is a considerable amount of research on spatial relations based representation and retrieval. Some of them are good for pictorial matching instead of similarity retrieval such as 2D *-string series ([Chan97], [Chan91b], [Lee91], and [Wu94] etc.), 9DLT approach ([Chan91a]), and SHV approach ([Zhou97]). We are not going to compare them here. [Hern94]'s approach does not have a similarity measure and we have made some comparison on the representation aspect in the previous discussion. Hernandez's work is mainly on qualitative spatial relation representation and reasoning that is more on spatial database area.

There are two other approaches that are quite close to ours in terms of using graphs to represent spatial relation for similarity retrieval. They are θ*R*-string

([Gudi95]) and 2D-PIR ([Nabi97]). However, in [Nabi97], Nabil has already done some comprehensive comparison with Gudivada's approach. Therefore, in the following, we only use 2D-PIR as a typical reference to compare with IO&T and AOSR in detail.

The experimental databases are generated automatically as illustrated in last chapter, i.e. each database consists of 600-700 pictures with each image contains 10 to 15 objects with extent. The pictures are divided into 5 groups according to the number of objects' changes (such as moving and rotating) with respect to the query picture. There are 20, 45, 120, 210, and 252 pictures in the group G1, G2, G3, G4, and G5 respectively. A picture is in Gi when there are i objects changing their positions with respect to the query image. Pictures with five or more objects changing their positions belong to G5.

For the first experiment, we randomly choose one unchanged object as a dominant object, and calculate for each group the average similarity degree. We normalize the maximum similarity degree to 100 for all three approaches, and use integer values for similarity only. The results are shown in Figure 5.6.

It is expected that 2D-PIR will have the highest values of similarity degree for groups with fewer objects (especially one object) change and IO&T the lowest. Since 2D-PIR is not capturing the rotation of objects, some changes might not be significant enough to change the respective 2D string that is based on MBRs. Therefore, pictures are still regarded as similar even after some changes. On the other hand, AOSR can detect the rotation of objects, but not the topological changes whereas IO&T can detect both. This does not mean IO&T is not efficient. On the contrary, it only says that IO&T's similarity measure has a greater discriminative power to uncover changes went unnoticed by other measures. In fact, the effectiveness of a similarity retrieval algorithm depends on the ranking values instead of the raw similarity values in an application. The following experiment shows that IO&T is good for the similarity retrieval ranking.

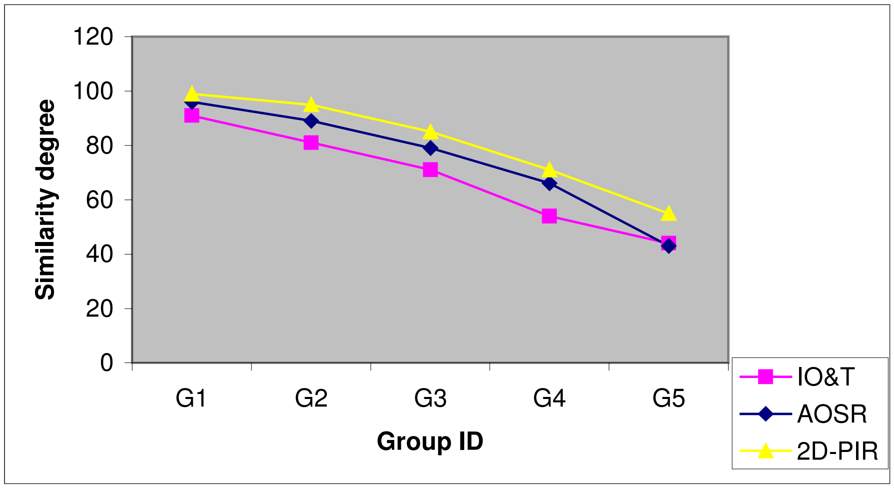


Figure 5.6. Objected-based retrieval experiment results

Table 5.1. Distribution of picture retrieved for IO&T

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	100.00	0.00	0.00	0.00	0.00
20	75.00	25.00	0.00	0.00	0.00
30	60.00	23.33	16.67	0.00	0.00
40	50.00	25.00	20.00	5.00	0.00

Table 5.2. Distribution of picture retrieved for AOSR

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	100.00	0.00	0.00	0.00	0.00
20	85.00	15.00	0.00	0.00	0.00
30	63.33	30.00	6.67	0.00	0.00
40	50.00	32.50	15.00	2.50	0.00

Table 5.3. Distribution of picture retrieved for 2D-PIR

Ranks	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group4	Group 5
10	100.00	0.00	0.00	0.00	0.00
20	90.00	10.00	0.00	0.00	0.00
30	63.33	23.33	3.33	0.00	0.00
40	50.00	47.50	2.50	0.00	0.00

For the second experiment, we calculate all S_1 to S_n and their total to get the top 10, the top 20, the top 30, and the top 40 pictures with higher similarity and compute the percentage distribution of images that have been retrieved from each group. Table 5.1, Table 5.2, and Table 5.3 show the results of the experiment for different approaches respectively.

From Table 5.1 to 5.3, we can see that the percentage of pictures retrieved from groups with fewer object changes is always higher for all approaches. This means all three approaches have some common standards about the similarity that relates directly to the number of objects' changes in a picture. However, when we rank more pictures for different approaches, there are more and more pictures being retrieved from groups other than the first group (in particular for IO&T and AOSR). This shows that even when many objects have shifted their positions, as long as their extents of changes remain small, they will be retrieved as pictures similar to the query pictures. However, this is not the case for 2D-PIR, whose results are more restricted to the first few groups (i.e. those groups with less number of objects involving movements. In this aspect, 2D-PIR is not as sensitive as IO&T and AOSR to detect the extent changes. This could be due to the fact that the representation of 2D-PIR has the duplication of 2D string part and topological relation part. The representation approach itself has its weakness in the accuracy though the similarity measures used are the same (i.e. degree similarity measure). On the other hand, AOSR is also not as sensitive as IO&T to detect the extent changes based on the results. This is expected since we have not considered topological part in AOSR. As we stand, a good representation approach should be able to detect object movement as well as extent changes. With a very small number of objects involving movement, the picture tends to have a high value in similarity. However, when there are few more objects involving changes, the extent changes should become significant to affect the similarity. This is why we consider the result of IO&T the best among the three.

However, AOSR does have its advantages over IO&T in some application areas. Typically, these applications do not consider topological relations. For example, some applications consider two touched objects as one object (such as a face recognition system that bases on the spatial relations of eyes, nose, and mouth, etc.). In such a case, IO&T is not as good as AOSR to detect just non-topological extent that includes distance and extension etc. Therefore, in applications involving both topological relation and orientation relation, IO&T is preferred over AOSR. On the other hand, in applications involving non-topological spatial relation, AOSR is better than IO&T. It is also possible to integrate both AOSR and IO&T in some applications that need more elaborated representations for similarity retrieval. Although we are not going to discuss all the detail about how/when to use both representations in different scenarios, they do have the flexibility and adaptability based on their nature of objectiveness and accuracy as shown in the discussions and experiments.

5.4 Summary

In this chapter, we have discussed the augmented orientation spatial relation (AOSR) representation to include the extent of objects and the similarity retrieval algorithm based on it. Compared with the existing systems, the proposed approach is not only rotation invariant, it is also capable to capture the relative distance and orientation range between objects. It overcomes the ambiguity problems that exist in other orientation representations, and is more flexible and applicable.

Note that we only focus on orientation spatial relation part for AOSR. The topological information is not covered by AOSR. However, AOSR is orthogonal to other topological spatial relation representations (such as [Egen91]). Therefore, we may simply combine our proposed AOSR with Egenhofer's topological representation if we need to include topological information.

In addition, from the experiments based on IO&T, AOSR, and 2D-PIR, we see that two typical factors are affecting the similarity perception. One is the number of objects that have changed their spatial relations, and the other is the extent of these changes. Different approaches have different emphasis on these two factors though a picture with many objects changed their positions will be more likely rejected in most of similarity retrievals. In real life, it is arguable if a small extent of changes is considered as significant as the spatial relation changes of objects. This is the reason why we introduced different approaches that can complement each other on this point.

Chapter 6

Towards interactive similarity retrieval process

6.1 What is interactive retrieval

We have discussed spatial relation representation, similarity metrics, and similarity retrieval so far. Most picture retrieval systems discussed in previous chapters emphasize on spatial relation representation with a metric system. There are some general discussions about this kind of metrics system in [Elkw99], [Elkw00], [Mao03] etc. All these systems use a measure to cluster or rank pictures from the pictorial databases.

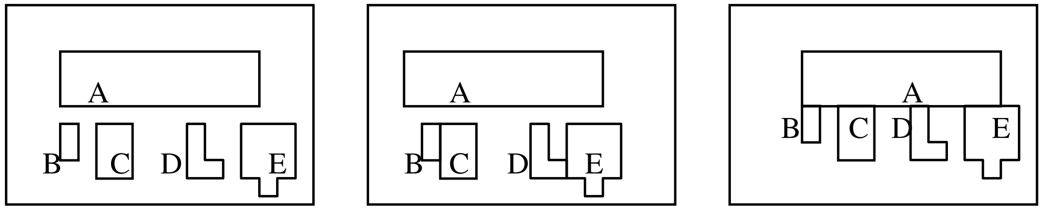
The clustering approach (such as [Wu01] and [Li02]) has a problem in handling images near the boundary of clusters. Suppose there are two clusters in a database, picture A and picture B in cluster one, and picture C in cluster two. When B and C are at the boundary of their clusters whilst A is in the middle of its cluster, it is possible for B to be nearer to C instead of A. Therefore, the data distribution does not always favor clustering approach.

On the other hand, when pictures are ranked (such as [Ang98] and [Nabi96]), the wanted picture may be ranked out according to the measure computed by the metrics. This does not mean the metrics is not good. In fact, the key reason is that the system measure is fixed whilst the user's perception of similarity is subjective.

At this stage, it is hard to find a system that guarantees the user to find the expected picture directly. Here are some factors causing the difficulty:

- Psychological and physiological factors: The similarity issue is inherently subjective. It varies from one user to another depending on the context at times. However, a pre-defined measure system is objective and independent of the context. Therefore, the retrieval is not merely a matter of finding a good measure, but of finding an effective way that is adaptable to understanding and reflecting user needs as well.
- The gap between the abstract representation and the real image: Unlike textual query, it is difficult to provide an exact description about a real image, and there is no perfect metric system that will consider all details of images. We need to find ways to fill the gap in order to meet the user's needs.

It is too challenging to define such a perfect system metric to satisfy different purposes and preferences from different users. Let us re-look at the same example in chapter 4 as follows.



(a) Q (b) P1 (c) P2

Figure 6.1. An example from chapter 4

In Figure 6.1, we have 3 pictures Q, P1, and P2. Suppose we consider spatial relations only, and we want to ask the question: *Which picture between P1 and P2 is more similar to Q?* We might have a unique for this question if we use only a pre-defined metric. However, in real life, we do have different answers from different readers. This is because each reader will use a personal metric to measure the similarity. Although human have some common standard about similarity, the fuzzy area for individual perception is undeniable. Therefore, when we develop a similarity retrieval system, we might base on the general common standards to index the pictorial database, and at the same time, we want the user to decide which pictures are more similar based on his/her personal metrics during the retrieval process. This is called interactive similarity retrieval [Bart01], [Kim03], [Rui98]. For the interactive retrieval, in the worst case, we may have to examine each candidate picture in the database until the most wanted picture is found.

6.2 Feedback based similarity retrieval

There are many distance measures for spatial relationship similarity. A typical similarity distance between a query picture Q and a database picture P can be defined as:

$$\mathbf{D_{SIM}(Q, P) = \sum_{j=1}^{m} W_j\, D_j(Q, P)} \qquad \mathbf{(6.1)}$$

where D_j is the similarity distance in dimension j, m is the dimension of the space to be measured, W_j is the weight of the similarity distance D_j for dimension j contributing to the total similarity distance D_{SIM}. However, the choice of W_j is rather subjective. A typical example is the IO&T metric that includes topological dimension and orientation dimension. In the previous chapter, we set all the weights equal in the system measure. However, it is possible to adjust the weight for different similarity dimension. The naïve process to adjust W_j has three steps:

Step 1: The user inputs some feedback based on the candidate pictures returned from the system.

Step 2: The similarity system updates W_j.

Step 3: The similarity system recalculates the similarity distance based on the new formula and return new candidate pictures

These steps are repeated until the target picture is identified. The calculation in Step 3 for every database picture is intensive. This is because we have no ideal index for interactive similarity retrieval. In the following discussion, we first introduce a digraph index structure that can be used to navigate within an image database to according to user feedbacks.

6.3 Building a k-regular digraph index

A k-regular digraph index is a digraph of fixed out-degree k. Suppose we want to display only the three most similarity pictures based on a selected similarity metric in a database with only 7 pictures A to G. The graph in Figure 6.2 is a 3-regular digraph showing each picture with links to its three most similar pictures. It is implemented in Figure 6.3 as an adjacency list, where A, B, C, D, E, F are image ids, numbers before the respective arrow are distances (weights of the pointers) between the two images connected by the arrow.

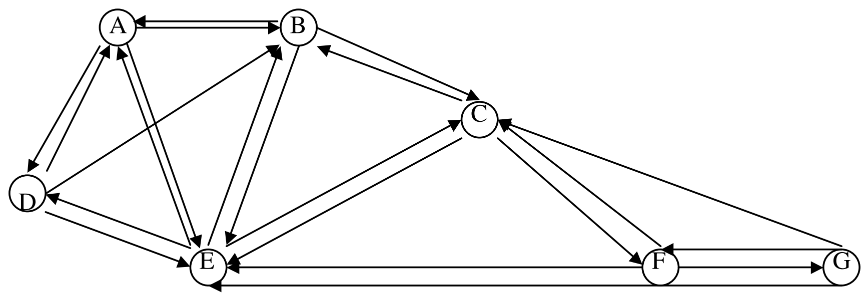


Figure 6.2. Digraph of 3 most similar pictures

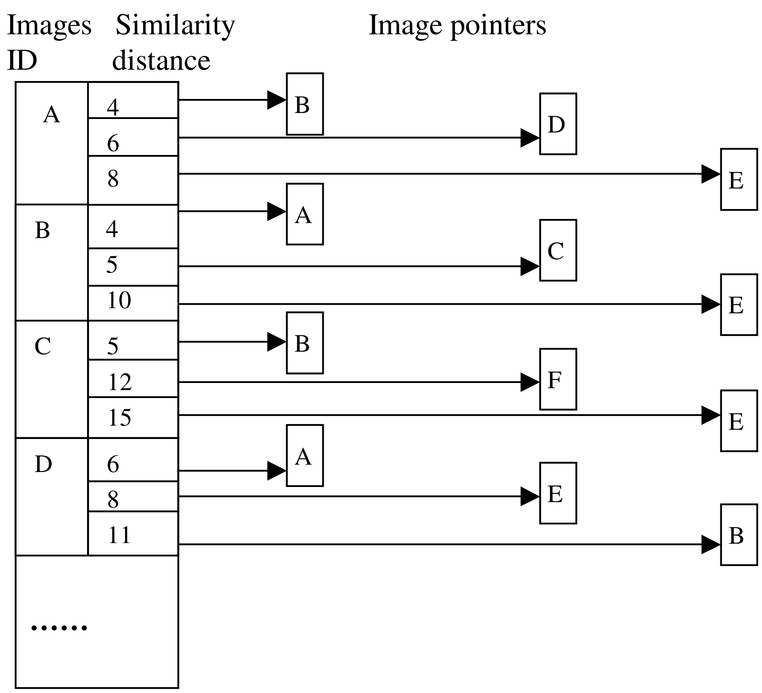


Figure 6.3. Implementation of the digraph

To build a k-regular digraph, we repeatedly insert a new image using the following algorithm. For the ease of discussion, k is assumed to be 3.

Algorithm 6.1: Insert a new picture into a 3-regular digraph

Input: A new picture Q, a 3-regular digraph

Output: The updated 3-regular digraph

begin

1. Initialize all three pointers of the new picture Q to NULL, and set respective 3 distance weights Q[1], Q[2], Q[3] to maximum.

2. For a picture P in the database, calculate the similarity distance DP with the new picture based on a selected metrics system.

3. Compare the distance with the three most similar pictures of P. Suppose the three weights are P[1], P[2], P[3].

  a. *if DP < P[1], set P[3] = P[2], P[2] = P[1], P[1] = DP*
  b. *else if P[1] <= DP < P[2], set P[3] = P[2], P[2] = DP*
  c. *else if P[2] <= DP < P[3], set P[3] = DP*

4. For the new picture, reset its pointer as Step 3 as well.

5. Go to the next picture in the pictorial database, and go back to Step 2.

end

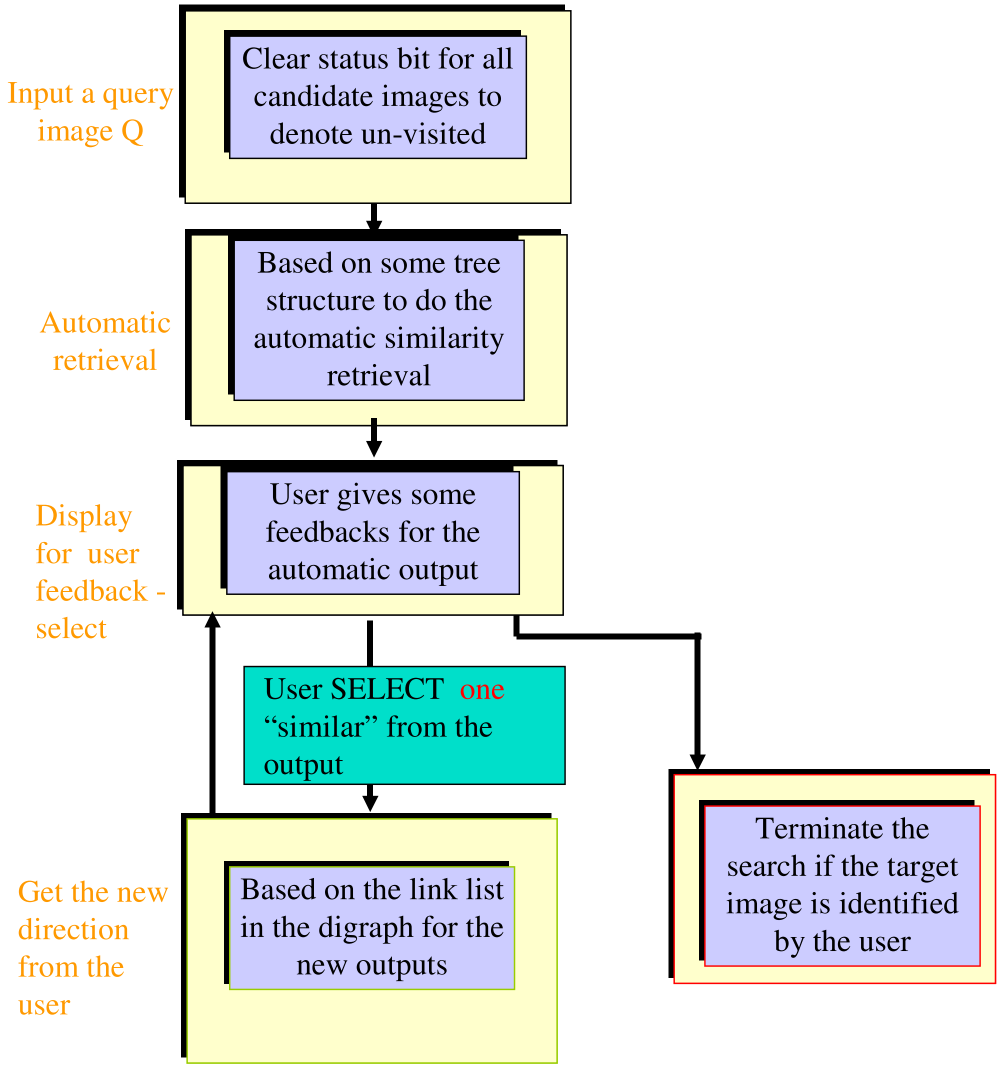
The complexity of this algorithm is O(n) where n is the number of pictures in the database.

This digraph index structure is at the bottom of some typical tree index for the automatic retrieval discussed in previous chapters. The array of index Ids are actually the leaf nodes of the traditional tree index built for the system measure. In the next section, we will see how to make use of the digraph for the interactive navigation to complement the automatic similarity retrieval.

6.4 Interactive similarity retrieval based on k-regular digraph index

A k-regular digraph of a pictorial database is needed for the ease of interactive similarity retrieval/navigation [Lee04]. Typically, after the automatic similarity retrieval, the user may not be able to find the target picture based on the first return from the system. Instead of going back to have another search in different part of the database, the user might want search forward from the subset of pictures returned. It is believed that a good system should find some approximate candidates based on the pre-defined system measure. Therefore, the final target picture should be near to what the system returns in the database space. The user will be able to find the target picture from the given approximate candidates.

The digraph is useful for this interactive process. In the following retrieval algorithm, given an iconic query picture as input, we use a temporary array to contain the overall similarity ranking based on a predefined metrics system. For each database picture, there is a status bit to indicate if it has been visited during the retrieval process. The interactive retrieval process flow and the detail algorithm are illustrated as follow:



Algorithm 6.2: Interactive Similarity Retrieval

Input: A query picture and a 3-regular digraph

Output: Candidate picture

begin

1. Set all pictures in the picture database as un-retrieved;

2. Calculate the similarity distance DP between the query picture and every picture in the database based on a selected metrics system. The distance values will be sorted and stored in an array AR with picture id.

3. Display the top three similar pictures to the user.

4. User are supposed to input:

a. If the candidate picture found, return;

b. If no candidate picture, but the most similar one P is selected among the three returned pictures, then go to Step 5.

5. Mark the top three pictures as retrieved, and display the next three pictures based on the following rules:

a. If all three pictures associated with P have not been retrieved yet, display these three pictures, and then go to Step 4;

b. If there are pictures marked as retrieved among the three pictures pointed by picture P, we pick some un-retrieved pictures from array AR to display as the next three pictures, and then go back to Step 4.

end

Note that the interactive process takes place after the ranking of candidate pictures based on the system similarity metrics. Users will guide the retrieval process through feedback. When a user rejects the next three nearest pictures, the system will go back to the ranking array to get other candidate pictures in the waiting. This avoids two typical issues of interactive retrieval algorithms:

1. The retrieval process will not go into an endless loop in navigation.
2. This will ensure that pictures in different connected components will be scanned eventually.

For example, Figure 6.4 shows the graph associated with an iconic image database. There are two clusters of pictures, and the pictures in different cluster are not linked. When the candidate images are all from one cluster:

- We may get into infinite loop if we never mark a visited picture;

• We will never get to the pictures in the other component if we could not pick a picture from the other component when the first component is exhausted.

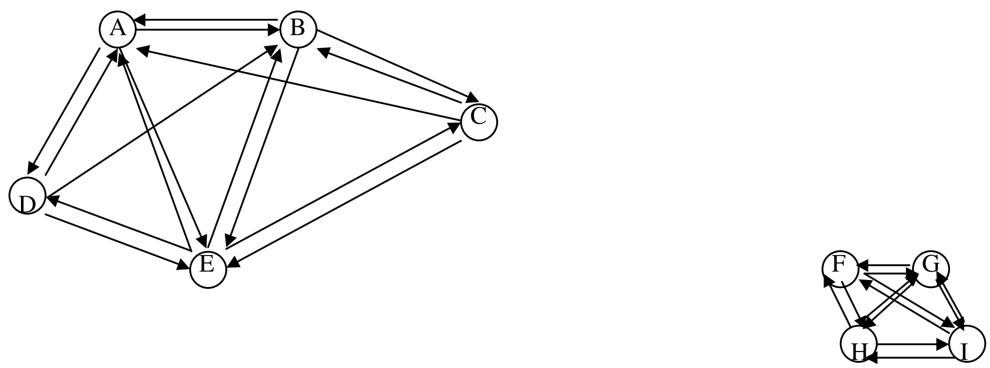


Figure 6.4. A digraph with 2 components

In general, the new picture similarity retrieval process is an exploration of the database based on the directed graph. The objective of this approach is to avoid the complicated recalculation of the similarity degree. The complexity of retrieval is O(n). However, having to scan the table is unusual, and the performance should be much better in a real situation. The complexity of building the digraph for a picture database with n pictures is O(n^2). This heavy work is done at the preprocessing/build stage rather than the retrieval stage as compared to other interactive retrieval approaches [Pork99] [Liu06]. This will certainly improve the performance of the retrieval process significantly.

The limitation of this approach is that the strict fixed bandwidth (number of pointers) of this index structure has some impact to the retrieval and update of the picture database.

• When we increase the bandwidth, the possible number of clusters in the picture database will decrease.

- The bigger bandwidth means more pointers need to be updated when a picture is added.
- The bandwidth is related to the number of candidate pictures returned to user at one time.

In the following discussion, we will introduce some enhancement for the digraph structure.

6.5 Enhancing a k-regular digraph

In the previous sections, we discussed the use of a k-regular digraph for interactive retrieval. A k-regular digraph is a directed graph with fixed out-degree. For a 3-regular digraph, since we must have each image point to 3 most similar images, we have to make a choice if there are more than 3 images having the same similarity distance from the target image. For instance, in Figure 6.5, although A, B, C and D are similar to E, only A, B, D are chosen, and hence there is no pointer from E to C. On the other hand, even though G is quite far away from E in similarity, G has a pointer pointing to E as we need 3 pointers for G as well. These pointers may be misleading sometime when we follow them to navigate in the database. Therefore, we would like to enhance the k-regular digraph by using variable number of pointers for navigation. The new structure will be called an enhanced digraph.

The implementation of an enhanced digraph index structure is illustrated in Figure 6.6. In general, an enhanced digraph uses a list of variable number of pointers instead of fixed number of pointers for each iconic image in the digraph index structure.

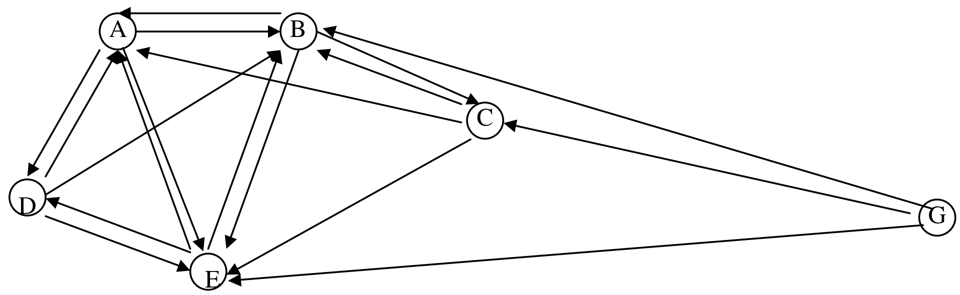


Figure 6.5. Graph of an example image database

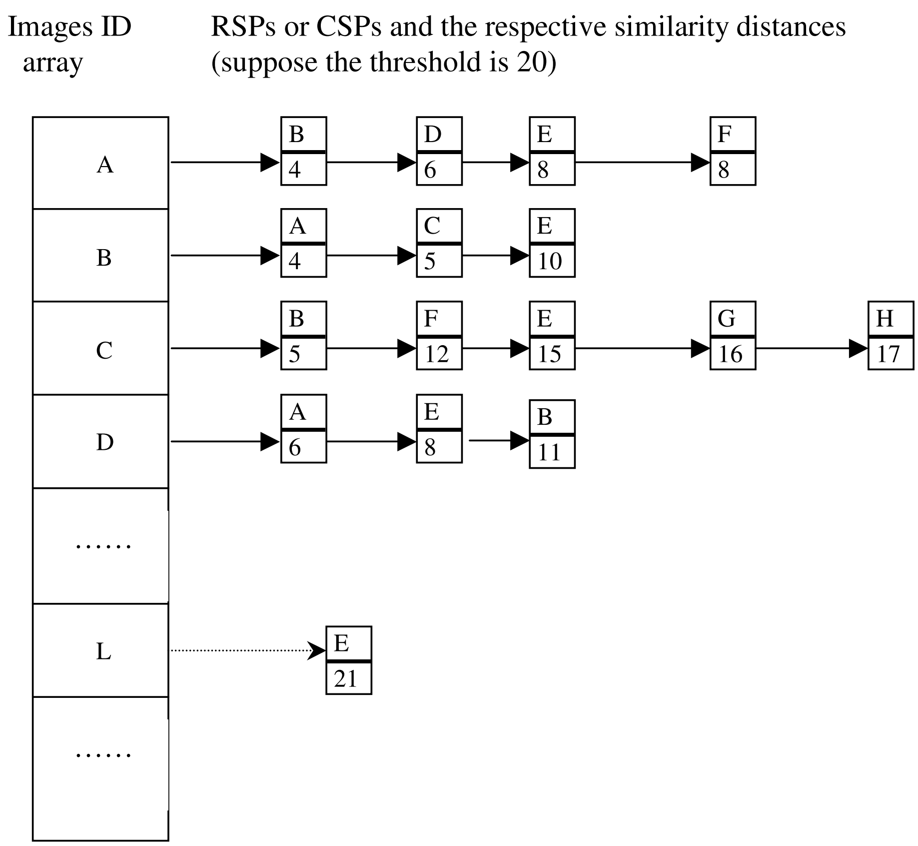


Figure 6.6. An enhanced digraph

Two different kinds of pointer are introduced in the enhanced digraph based on a pre-defined threshold on similarity distance for each image:

• Regular Similarity Pointer (**RSP**), denoted as

For each image in a database, there are m linked RSPs pointing to the m most similar images in the database where m is a variable. All these m images are with similarity distances less than a pre-defined threshold value.

• Connectional Similarity Pointer (**CSP**), denoted as

When there is no RSP for an image in a database, a compulsory pointer is needed to connect it to the nearest image even if the distance is bigger than the upper threshold value. This is to avoid too frequent recursive call during the retrieval.

With the enhanced digraph, we can have a variable number of images linked from each image. The threshold can be adjusted according to the need. The objective of the adjustment is to control the number of pointers without the loss of the flexibility.

The algorithm below is used to build the enhanced digraph.

Algorithm 6.3: Insert a new image into an enhanced digraph index structure

Input: A new image Q and an enhanced digraph index structure

a similarity measure,

and a threshold value θ

Output: Enhanced digraph with the new image inserted

begin

1. Initialize the RSP and CSP pointers' header of the new image Q to NULL and set respective distance weight to maximum.

2. For each image P in the database, calculate the similarity distance DP with the new image based on the input measure.

3. Compare the distance DP with the threshold θ:

a. if the similarity distance is bigger than the threshold,

• For the image P, if there is no RSP pointer, and if the similarity distance DP is smaller than the weight of CSP pointer of the image P, make CSP point to Q, reset the CSP weight to the similarity distance DP.

• For the input new image Q, if there is no RSP pointer, and if the similarity distance DP is smaller than the weight of CSP pointer of Q, make CSP point to P, reset the CSP weight to the similarity distance DP.

b. if the similarity distance is smaller than the threshold θ, set both CSP pointers of Q and P to NULL,

• Insert a new pointer in the linked list of RSP pointers of P.

• Insert a new pointer in the linked list of RSP of Q.

4. Go back to Step 2 for the next image in the image database, if exists, else stop.

end

Note that we have the CSP only when we have no RSP. With these pointers, we can do vertical, horizontal and recursive (VHR) retrieval, by branching out from a linked list of RSP, moving along the linked list, and resuming from where it left after a vertical movement respectively. As mentioned early, the candidate set (mentioned in the algorithm) is obtained (i.e. the first set of candidates) from an automatic index or clustering approaches (such as M-tree [Ciac97]).

Algorithm 6.4: VHR Similarity Retrieval

Input: A query image and an enhanced digraph

Output: Candidate image

begin

1. Set all candidate images as un-retrieved.

2. For the given query image, calculate the similarity distance DP with every image in the candidate set returned from an automatic retrieval method. Sort the distance values and store them in an array AR with the respective image ids.

3. Display the top k most similar images to the user.

4. User has the following options:

a. The target image is found, SELECT the target image, stop.

b. No target image found, but choose one most similar image P among the returned images. Then follow the link of the most similar image P. Go to Step 5 (vertical retrieval).

c. No target image found, also no similar candidate chosen. Click on the next link for the system output. Go to step 6 (horizontal retrieval).

5. Push the next unvisited pointer for the current link list onto the stack. Mark all current images as retrieved, and display the next images based on the following rules:

- *If P has more than k unvisited RSP, display these images based on the order, and then go to Step 4;*
- *If P has less than k but more than 0 unvisited RSP, display all of them, and then go to Step 4;*
- *If P has only the CSP, display the image through CSP, and then go to Step 4;*
- *If P has no unvisited RSP, and has no CSP, then go to Step 7.*

6. If there are still un-visited RSP pointers for the current link list, mark all current displayed images as retrieved, display up to k unvisited images pointed by RSP pointers from the current link list, go back to Step 4; Else (i.e. no unvisited RSP points) go to Step 7.

7. Pop out one pointer from the stack if it is not empty, go to Step 6 (recursive retrieval). If there is no more unvisited image, STOP. Otherwise, display up to k unvisited image from the array AR.

end

In general, the user will guide the retrieval process through feedback using the enhanced digraph. When the feedback from the user leads to a dead end, the system will lead the user back to the stack of pointers to continue the disrupted retrieval process.

6.6 The coverage experiments

Similarity retrieval is fuzzy and subjective. There is no widely accepted benchmark for similarity retrieval experiments. In particular, it is extremely subjective to design an interactive experiment to demonstrate the effectiveness of a similarity retrieval algorithm. The objective of the coverage experiment is to show that the proposed digraph index for the interactive retrieval do not miss any important part of the search space even without re-computation of similarity measure. This is more objective than the survey style experiments involving a small number of subjective testers.

The experiment is setup in the same way described in the previous chapters. In short, there are 647 pictures in our database. Each image contains 10 to 15 objects. The pictures are divided into 5 groups according to the number of objects' changes (include translation, rotation, etc.) with respective to the query picture. There are 20, 45, 120, 210, and 252 pictures in Group 1, 2, 3, 4, and 5 respectively. A picture is in Group i when there are i objects changing their positions with respect to the query image. Those pictures with five or more objects changing their positions belong to group 5.

We use IO&T metric to build the index structure of the database. We have built 2 indexes, one fixed at 5 pointers and the other is an enhanced digraph. The retrieval processes based on the 2 indexes are denoted as Pidx5, and PidxΔ. The retrieval process without using any index is called Pidx0. Since Pidx5 is good for displaying 5 pictures each time, we will display 5 candidate pictures for Pidx0 and PidxΔ as well. The pictures reachable in each loop for Pidx0 are: the top 5, the top 10, then the top 15 and so on. However, for Pidx5, we can reach the top 5, minimum 10 to maximum 5 x 5 + 5 possible candidate pictures, minimum 15 to maximum 5 x 5 x 5 + 5 x 5 + 5 possible candidate pictures, and so on. We calculate the possible covered candidate pictures distribution among all groups after any possible feedback from the user. Note that we have not used any clustering or filtering approaches in the experiment since it is more objective to compare the results with the retrieval without using any index.

Table 6.1. The candidate distribution for Pidx0 with 5 pictures display each time

The possible candidates covered after each loop of the return	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group 4	Group 5
5 (the automatic return)	100.00	0.00	0.00	0.00	0.00
10 (after the 1st feedback)	100.00	0.00	0.00	0.00	0.00
15 (after the 2nd feedback)	93.33	6.67	0.00	0.00	0.00
20 (after the 3rd feedback)	85.00	15.00	0.00	0.00	0.00

Table 6.2. The candidate distribution for Pidx5 with 5 pictures display each time

The possible candidates covered after each loop of the return	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group 4	Group 5
5 (the automatic return)	100.00	0.00	0.00	0.00	0.00
15 (after the 1st feedback)	86.67	13.33	0.00	0.00	0.00
36 (after the 2nd feedback)	50.00	47.22	2.78	0.00	0.00
58 (after the 3rd feedback)	34.48	51.72	12.07	1.72	0.00

Table 6.3. The candidate distribution for PidxΔ with 5 pictures display each time

The possible candidates covered after each loop of the return	The distribution percentage among groups				
	Group 1	Group 2	Group 3	Group 4	Group 5
5 (the automatic return)	100.00	0.00	0.00	0.00	0.00
13 (after the 1st feedback)	92.31	7.69	0.00	0.00	0.00
33 (after the 2nd feedback)	60.61	33.33	6.06	0.00	0.00
50 (after the 3rd feedback)	40.00	35.20	22.80	2.00	0.00

Now we compare and analyse the above result in detail as follows:

In table 6.1, we have no index for interactive retrieval. Therefore, if we are allowed to display 5 pictures each time, the possible candidate pictures will be the top 5, the top 10, and the top 15, and so on based on IO&T metric. Initially, the candidates returned are mostly from group 1 since there are very few object changes in Group 1. This means the metric IO&T is effective as a common similarity measure. However, if more and more pictures are returned, there will be some pictures from other groups retrieved instead of all 20 pictures in Group 1 retrieved first. For example, after the third feedback, the top 20 pictures based on IO&T will be returned. However, there are only 85% from group 1, which are 17 pictures. The other three pictures in Group 1 are not covered by the top 20 returns. If the target picture happens to be in the remaining 3 of group 1, the user may need a few more iterations in order to reach the target picture.

Table 6.2 is used for Pidx5. The first 5 pictures returned by the automatic retrieval are the same as Pidx0 since they are using the same base metric. However, after one feedback from the user, there are maximum 5 x 5 candidate pictures linked from the first 5 pictures. Therefore, there should be maximum 5 x 5 + 5 possible candidates being covered (reachable). However, the experiment result shows that only 15 pictures are covered after any one feedback from the user. This is because some pictures have high in degree as they are pointed to from many other pictures. In spite of the overlap of pointers, the number of pictures reachable in Pidx5 is still bigger than that of Pidx0 after the same number of feedbacks from the user. Hence, it makes sense that more pictures from each group will be covered as possible candidates. In fact, after the third feedback, all 20 pictures from group 1 are reachable by the user (58 x 34.68% = 20).

In table 6.3, we use the enhanced digraph. It can be seen that the number of pictures reached after each feedback is less than the number for Pidx5 though more than the number for Pidx0. This is because we no longer use the fixed out-degree for the digraph. It is possible that there are less than 5 pictures displayed sometimes. From the tables, we can see that the number of possible candidates covered after each loop of the return using PidxΔ is more than the number using Pidx0 though it is slightly smaller the number using Pidx5, and also the distribution percentage is better (i.e. favor groups with less object changes) for PidxΔ comparing to that for Pidx5. Therefore, in general, the enhanced digraph index has retained the advantage of Pidx5 of wider coverage. At the same time, it overcomes some drawbacks of the fixed out-degree indexing structure.

The experiment results show clearly that the user can reach more candidate pictures with fewer feedbacks when an index is used. In particular, the pictures with fewer changes can be fully covered as candidate pictures within a few feedbacks. This shows that the approach using the digraph with user feedback for similarity retrieval will not lose any possible candidates coverage even with less re-computation.

6.6 Summary

Up to now, we have discussed the interactive similarity retrieval based on the digraph index structure as illustrated in Figure 6.7. The approach relies on the interaction with the users to navigate through the index structure. The user's selection will be used as the new query point for the next cycle of outputs. Since we have linked the nearest neighbors for each point in the search space, the re-search space will be reduced and the re-computation will also be saved. From the coverage experiment, we can find that we won't lose coverage of possible candidates though the search space is reduced for individual iteration. However, one of the problems is that a user may not be able to give a clear single feedback always. In practice, a user may select multiple points as the relevant feedback in some iterations. In that case, we need to either generate a new representative query point based on the feedback points or use multi-point query. Generating a representative query point based on the feedback points may not work since the new point may not be in the search space (database) initially. Therefore, we will discuss what is multi-point query and how to use the proposed digraph for multi-point query in the following chapter.

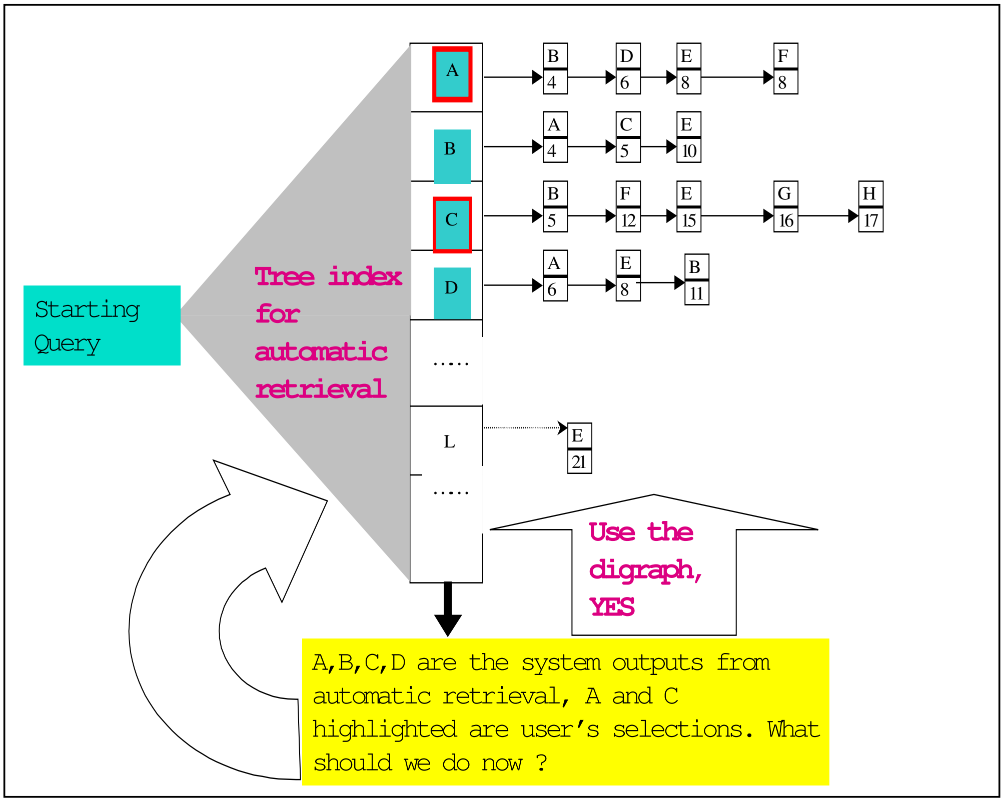


Figure 6.7. The interactive retrieval based on digraph

Chapter 7

Using multipoint feedback in interactive similarity retrieval

In Chapter 6, we have discussed the digraph structure for the interactive similarity indexing and retrieval. In this chapter, we are going to discuss some interesting heuristics based on multipoint feedback: dynamic re-weighting metric, and feedback adjustable index for the interactive retrieval.

7.1 Multipoint model for an interactive similarity retrieval system

In an interactive similarity retrieval system, a user poses an example query picture Q that is similar to the target picture in a database. Suppose we consider the database as a space while each picture as a point, we need to navigate in the space based on the given query point(s). We refer to Q as the “starting” query in this case. A special case for this model is that the user is allowed to modify the Q interactively to reset one new "starting" query. However, a more general case is that the modified query points can be more than one.

Definition 7.1 (Multipoint Query) A multipoint query Q = (n, P, WP, D) for a database space consists of the following information:

1. The number n is the size of the query Q.
2. A set of n points $P = \{P_1,\ldots,P_n\}$ in the database space.
3. A set of n weights $WP = \{WP_1,\ldots,WP_n\}$, the ith weight WP_i being associated with the ith point P_i.
4. A distance function D.

For the starting query, we have n = 1 which is a special case. After the automatic similarity retrieval based on the first starting query, there can be more than one return that look similar to the target picture. In such a case, the user should be allowed to select all or some of the similar pictures. The aggregated multiple feedbacks is called a multipoint query.

In some existing approaches, such as [Ishi98], the user is supposed to adjust the respective weight (scores) for each new feedback query point according to the ranking/degree of similarity. They call this "goodness" scores for each feedback point. However, this kind of adjustment is too subjective and very fuzzy for an un-trained user since different users may have different sense of scoring. Therefore, we will typically assign all $WP_i = 1/n$. In this case, the similarity distance between a database-picture DP and the multipoint query is defined as:

$$\mathbf{D(Q, DP) = \sum_{i=1}^{n} D(P_i, DP)/n} \qquad \mathbf{(7.1)}$$

Note that the computation involves multiple pictures and may include multiple spatial features (such as topological, orientation, x-axis/y-axis in IO&T and 2D-PIR etc) in the computation of each $D(P_i, DP)$. When n = 1, D(Q, DP) = D(P, DP), and we have the original distance function.

The multipoint feedback has richer information when compared to single point feedback as illustrated in Figure 7.1. Suppose we consider only two dimensions, i.e. topological relation and orientation relation. Then different hidden information can be extracted from different multiple point feedback. In the following, we will discuss how the digraph can be used for the general multipoint query retrieval.

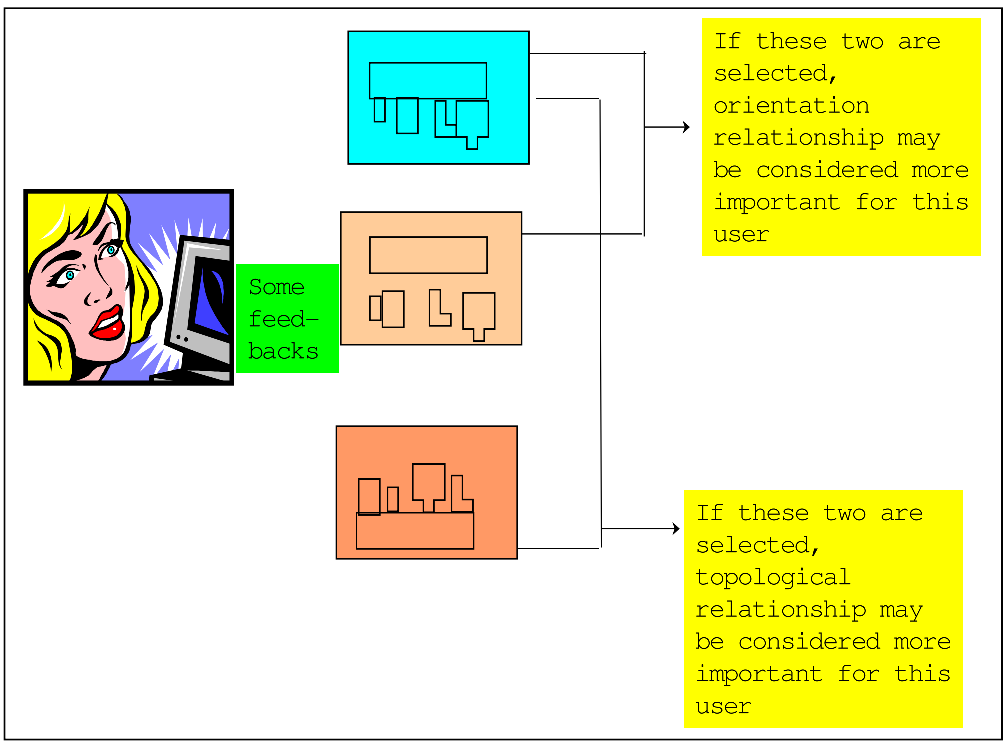


Figure 7.1. Hidden information derived from a multi-point feedback

7.2 Using digraph for the multipoint query retrieval

Before we discuss the retrieval algorithm, we review the distance measures for spatial relation similarity first. A typical similarity distance between a query picture Q and a database-picture DP can be defined as:

$$\mathbf{D_{SIM}(Q, DP) = \sum_{j=1}^{m} W_j D_j(Q, DP)} \qquad \mathbf{(7.2)}$$

where:

D_j is the similarity distance in dimension j,

W_j is the weight of the similarity distance D_j for dimension j contributing to the total similarity distance D_{SIM}.

In section 7.1, we have introduced the multipoint feedback, and formula (7.1) is used to compute the similarity distance for the multipoint query. If we combine formula (7.1) and (7.2), we should have the complete similarity distance of all dimensions for a multipoint query as follows:

$$\mathbf{D(Q, DP) = \sum_{i=1}^{n} (\sum_{j=1}^{m} W_j(D_j(P_i, DP)))/n} \qquad \mathbf{(7.3)}$$

where:

j denotes the similarity dimension,

i denotes the query point,

W_j denotes the weight for dimension j,

D_j is the similarity distance function for dimension j,

P_i means query point i,

DP is a database picture.

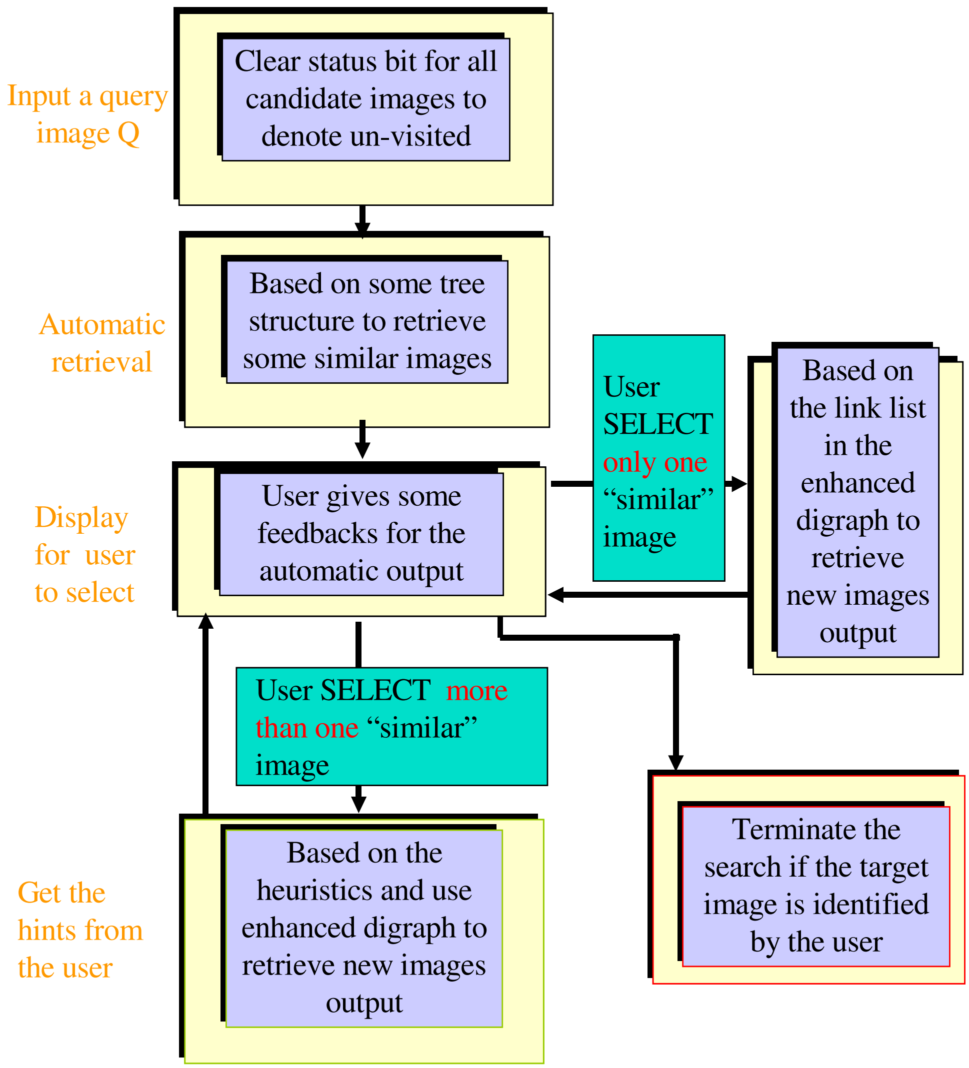
Since we fix the weight for each query point to be the same, 1/n is used as the weight for each query point.

However, in a multipoint feedback system, the weight for each similarity dimension is a factor that can vary and affect the outcome. In the following discussion, we will introduce a heuristic approach to make use of this weight information for each similarity dimension.

Definition 7.2: The **similarity distance range** on a dimension for a set of feedback-points is defined as the maximum similarity distance between any two feedback-points on that dimension.

Closest Neighbor Heuristic: If we have a smallest **similarity distance range** for the multiple feedback points from the user on one dimension, then the weight for this dimension should be maximized. The point with the smallest similarity distance from the previous point on this particular dimension is chosen as the next query point.

The idea is simple. If there are some common/similar parts/features among multiple feedback points, it is legitimate to assume that the user is looking for these features in the respective retrieval. Based on this heuristic, the similarity retrieval process can be adjusted as follow:



The following is the detail algorithm extended for multipoint query with the enhanced digraph.

Algorithm 7.1: **Interactive Similarity Retrieval with Closest Neighbor Heuristic**

Input: A query image and an enhanced digraph of a picture database

Output: Candidate image

Begin

1. Set all candidate images as un-retrieved;

2. For the given query image, calculate the similarity distance D with every image in the candidate set returned from the automatic retrieval. Sort the distance values and store them in an array AR with the respective image ids.

3. Display the top k most similar images to the user.

4. User has the following options:

a. The target image is found, SELECT the target image, stop.

b. No target image found, also no similar candidate chosen. Click the next link for the system output. Go to Step 6 (horizontal retrieval).

c. No target image found, but choose one most similar image P among the returned images. Then follow the link of the most similar image P. Go to Step 5 (vertical retrieval).

d. No target image found, but there are m > 1 similar images chosen. Calculate the similarity range for each dimension to find the next query point based on the closest neighbor heuristic. Go to Step 5 (vertical retrieval).

5. Push the next unvisited pointer from the current link list onto the stack. Mark all current images as visited, and display the next images based on the following rules:

- *If P has unvisited RSP, display these images based on the order, and then go to Step 4;*
- *If P has less than k but more than 0 unvisited RSP, display all of them, and then go to Step 4;*
- *If P has only the CSP, display the image through CSP, and then go to Step 4;*
- *If P has no unvisited RSP, and has no CSP, then go to Step 7.*

6. If there are still un-visited RSP pointers from the current link list, mark all current displayed images as visited, display up to k of them, go back to Step 4. Else (i.e. no unvisited RSP pointers) go to Step 7.

7. Pop out one pointer from the stack if it is not empty, go to Step 6 (recursive retrieval). If there is no more unvisited image, STOP. Otherwise, display up to k unvisited image from the array AR.

end

Note that we have not changed the distance functions even with multiple feedback points returned in this algorithm. There is no need to re-calculate the distances stored in the index structure during the preprocessing stage. The saving of distance computation makes the multipoint query processing very efficient. However, this approach might have missed some candidates since only one feedback point's link list is re-evaluated. Therefore, in the following, we discuss a dynamic weighting approach to cover more candidates with some limited re-computation.

7.3 Dynamic weighting based on the multipoint feedback query

In Section 7.1, we introduced the multipoint feedback concept. Formula (7.1) is used to compute the similarity for the multipoint query feedback. Ideally, if the user can accurately input the weight for each query point in a multipoint query (i.e. the similarity ranking or scores/degrees etc for each feedback), then the search can be steered towards the direction the user intended. However, an untrained user will have difficulty in providing a meaningful ranking. This is the reason why we will not adjust the weight for each query point.

In formula (7.2), there are weights for different dimensions of similarity. For example, in IO&T approach, we have weights for topological similarity and orientation similarity respectively. In a multipoint feedback system based on formula (7.3), the weight for each similarity dimension is adjustable based on the feedback. The following heuristic is proposed for this adjustment.

MinMax Heuristic: If we have a smallest **similarity distance range** for the multiple feedback points from the user on one dimension, then the weight for this particular dimension should be maximized in the new similarity measure. If we have a biggest **similarity distance range** for the multiple feedback points from the user on one dimension, the weight for this dimension should be minimized in the new similarity measure.

Using this heuristic, we adjust the W_j in formula (7.3) by observing that $\Sigma W_j = 1$. The revised weights apply to the computation needed in the next round of retrieval only. Suppose we have the biggest similarity distance range for dimension x, and the smallest distance range for dimension y, we will revise the related weights as follows:

- The new $W_y = W_{max} + (W_y / 2)$,
- The old $W_{max} = W_y / 2$,
- The new $W_x = W_{min} / 2$,
- The old $W_{min} = W_x + (W_{min} / 2)$,

where

$W_{max} = \max(\forall W_j)$, and $W_{min} = \min(\forall W_j)$.

This will reduce the weight for dimension x and increase the weight for dimension y in the new measure while maintaining the sum of all weights as unity.

This revision of weights makes the similarity metric dynamic and more accurate. The new adjusted metric will be applied to the candidate points linked to each feedback point. Since we have already seen in the previous experiment that the digraph have the reasonable coverage for the possible candidate set, it should be sufficient to use the approximate candidates linked from the multipoint feedback points in the digraph. The detail VHR similarity retrieval algorithm (Algorithm 6.4) is revised as follow:

Algorithm 7.2: VHR Similarity Retrieval with Dynamic Re-weighting

Input: A query image and an enhanced digraph of a picture database

Output: Candidate image

begin

1. Set all candidate images as un-retrieved;

2. For the given query image, calculate the similarity distance D with every image in the candidate set returned from the automatic retrieval. Sort the distance values and store them in an array AR with the respective image ids.

3. Display the top k most similar images to the user.

4. User has the following options:

- *a. The target image is found, SELECT the target image and stop.*
- *b. No target image found, also no similar candidate chosen. Click the next link for the system output. Go to Step 6 (horizontal retrieval).*
- *c. No target image found, but choose one most similar image P among the returned images. Then follow the link of the most similar image P. Go to Step 5 (vertical retrieval).*

d. No target image found, but there are m >1 similar images chosen. ***Go to Step 8.***

5. Push the next unvisited pointer for the current link list onto the stack. Mark all current images as retrieved, and display the next images based on the following rules:

- *If P has more than k unvisited RSP, display these images based on the order, and then go to Step 4;*
- *If P has less than k but more than 0 unvisited RSP, display all of them, and then go to Step 4;*
- *If P has only the CSP, display the image through CSP, and then go to Step 4;*
- *If P has no unvisited RSP, and has no CSP, then go to Step 7.*

6. If there are still un-visited RSP pointers for the current link list, mark all current displayed images as visited, display up to k of them, go back to Step 4. Else go to Step 7.

7. Pop out one pointer from the stack if it is not empty, go to Step 6 (recursive retrieval). If there is no more unvisited image, STOP. Otherwise, display up to k unvisited image from the array AR.

8. If there is no unvisited images linked from the m images, go to step 7, else adjust the formula of the similarity distance measure based on the m feedbacks and recalculate the new similarity distance D_{new} for all images pointed by these m images and mark the m images as visited. Go back to Step 3.

End

For this algorithm, the cost to adjust the weight is determined by the size of the link lists involved which is small as compared to O(N) (where N is the size of the database) without the digraph index. Since we can adjust the threshold to control the size of the link list, it makes sense to assume a constant C as the maximum size for each link list. Therefore, the complexity for each feedback cycle will become O(1) which is a big improvement in the performance.

7.4 Adjustable digraph index based on the feedbacks

In section 7.3, we have discussed dynamic similarity measure adjustment. However, we have also highlighted that we are not going to keep the new adjusted formula during each interactive process. Therefore, the discussed retrieval process based on the heuristics is not really a true "learning" system but a "learning" process for similarity retrieval.

Is it possible to retain some useful information from each retrieval to enhance the performance of future retrieval? Logically it seems possible though it will be too subjective to revise the similarity formula based on individual's retrieval process. In particular, when we have many pictures with a tied similarity distance in the indexing structure, it may be desirable to adjust the order based on the frequency of the selection from previous retrieval process. This adjustment will not affect the objectiveness of the system measure. We revise the VHR similarity retrieval process (Algorithm 7.2) to reflect this index adjustment as follow:

Algorithm 7.3: **VHR Similarity Retrieval with Dynamic Weighting and Indexing**

Input: A query image and an enhanced digraph index structure

Output: Candidate image and adjusted digraph index

begin

1. Set all candidate images as un-retrieved, and ***selection times to 0****;*

2. For the given query image, calculate the similarity distance DP with every image in the candidate set returned from the automatic retrieval. Sort the distance values and store them in an array AR with the respective image ids.

3. Display the k most similar images to the user.

4. User has the following options:

- *a. The target image is found, SELECT the target image,* ***and increase the selection times of the target image by 1. If the target image has the same similarity distance with the previous image in the respective link list, and selection times of the target image is bigger than that of the previous image as well, swap the position of the two images in the respective link list.*** *Stop.*
- *b. No target image found, also no similar candidate chosen. Click the next link for the system output. Go to Step 6 (horizontal retrieval).*
- *c. No target image found, but choose one most similar image P among the returned images,* ***increase the selection times of image P by 1. If image P has the same similarity distance with the previous image in the respective link list, and selection times of image P is bigger than that of the previous image as well, swap the position of the two images in the respective link list****. Then follow the link of the most similar image P among the k returned images. Go to Step 5 (vertical retrieval).*
- *d. No target image found, but there are m >1 similar images chosen. Go to Step 8.*

5. Push the next unvisited pointer for the current link list onto the stack. Mark all current images as retrieved, and display the next images based on the following rules:

- *If P has more than k unvisited RSP, display these images based on the order, and then go to Step 4;*

- *If P has less than k but more than 0 unvisited RSP, display all of them, and then go to Step 4;*
- *If P has only the CSP, display the image through CSP, and then go to Step 4;*
- *If P has no unvisited RSP, and has no CSP, then go to Step 7.*

6. If there are still un-visited RSP pointers for the current link list, mark all current displayed images as visited, display up to k of them, go back to Step 4. Else go to the next step.

7. Pop out one pointer from the stack if it is not empty, go to Step 6 (recursive retrieval). If there is no more unvisited image, STOP. Otherwise, display up to k unvisited image from the array AR.

8. If there is no unvisited images linked from the m images, go to step 7, else adjust the formula of the similarity distance measure based on the m feedbacks and recalculate the new similarity distance D_{new} for all images pointed by these m images and mark the m images as visited. Go back to Step 3.

end

There are not many changes in the algorithm except for the parts highlighted. A new status flag is used to denote the selection frequency of a picture, which is used to resolve the ambiguous tie issue. Note that we only keep information of those single point selections but not multiple point selections since the former is thought to give a clearer indication of the user's choice. The objective of this adjustment is to resolve the ambiguity. The multiple point selection itself shows that the users are not sure about the selection. Therefore, we will not take into account for this fuzzy information.

7.5 Related feedback based system and comparison

In Chapter 6 and Chapter 7, we have discussed feedback based similarity retrieval. In general, there are two types of relevance feedback systems. The first type is to capture a user's perceptual consistency in similarity retrieval for a long term. The system will be able to learn the regularities in a user's perception of similarity for the pictorial database retrieval in the target application domain. After learning, the database and the measure can be adjusted to favor the user's particular perception for a long-term usage. There are already numerous researches (such as [Agga01], [Bart01], and [Lim01] etc.) in this area that is like a data mining process. Typically, this kind of system needs a long period of training, and the database index or the similarity retrieval measure can be changed significantly after the training. It is used for a personalized system or a general output of an automatic system.

The other type of relevance feedback system is based on a short-term learning, i.e. its objective is to shorten the ad-hoc interactive similarity retrieval cycle (or session) by discovering the hidden subjective information from the user's feedback, and navigate efficiently in the search space (i.e. database). Since this hidden subjective information is not persistent and might be changed according to the context, the "learned" preference will not be kept for a long-term usage. This kind of system is also very practical in information retrieval. For example, when we use a search engine to find some wanted stuff from the inter-net, we may key in some keywords for the first search. If the target is not in the first return, we will either find a possible link to re-search or add more keywords etc. If the search engine can "understand" the intention of the user based on the feedbacks, it might help the user to find the wanted result quickly. Obviously, different users or a same user may have different intentions even when they input a same keyword at different time. The initial query is likely to be incomplete and not sufficient to describe the real target of the search from a user. Therefore, the feedbacks will be analyzed to find out the missing information. However, this kind of learned "intention" is per retrieval, and too subjective to be used for the next retrieval cycle (session). Our interactive research focuses on this ad-hoc "learning" system. This is new to iconic spatial similarity retrieval. However, there are some existing researches in the context of interactive text based retrieval [Rocc71], [Buck95], and the low-level feature-based image similarity retrieval [Ishi98], [Kim03], [Liu06], [Lu00], [Pork99], [Rui98], and [Wu00] etc. These existing approaches can be categorized into some typical types as follows:

Query point movement approach

Query point movement method (such as [Buck95] and [Ishi98]) is one of typical approach in making use of relevance feedback. For example, in [Buck95], Rocchio's formula [Rocc71] is used to compute the new query point Q^{new} from the old query point Q^{old} based on user feedbacks. In detail, the formula is expressed as follow:

$$\mathbf{Q^{new} = \alpha Q^{old} + \beta(\sum_{i \in D_R} D_i / N_R) - \gamma(\sum_{i \in D_N} D_i / N_N)} \qquad \mathbf{(7.4)}$$

where α, β and γ are suitable constants; D_R and D_N are sets of relevant documents and non-relevant documents respectively; N_R and N_N are the number of documents in D_R and D_N respectively. This approach is different from our approach. First, in [Buck95], a user is supposed to rank the candidates from highly relevant to non-relevant. This is very difficult and inaccurate for an untrained user to give ranking feedback without any guidance. On the other hand, user ranking is not expected in our system. Second, based on Rocchio's formula, the existing system can always add new keyword (or feature), and remove keyword (or feature) by giving a non-zero weight and a zero weight respectively. Therefore, the number of keywords (features) can change during a retrieval cycle. However, for our system, the feature space is fixed though the weights can be changed. Lastly, the approach mentioned above does not navigate within the reduced search space during a retrieval cycle which is the main difference from our approach. There is no combination of indexing the database for reducing the search space during the interactive cycle and adjusting the weight from query point movement approach as well. Typically, the initial outputs are produced by a random scan of the database, and then the system outputs the next query point after some extensive computation involving weights specified by the users.

Multi-point re-weighting approach

Another typical approach using relevance feedback is multi-point re-weighting [Rui98]. [Rui98] is based on the feedback scores (i.e. highly relevant, relevant, no-opinion, non-relevant, and highly non-relevant) to update multi-point weights dynamically instead of using Rocchio's formula to generate a new query point. The new search is to find the similar data points that have the smallest similarity distance to the multiple feedback points. In terms of heuristic weight updating, [Rui98] has some common attributes with our approach. However, [Rui98] relies on the explicit feedback score that is subjective and not really reliable for un-trained users. On the other hand, there is no mention about how to index and reduce the search space in [Rui98] as well. [Lu98] is another similar work that integrated semantic information into the re-weighting and query point movement, which has the similar nature of [Rui98] and [Buck95].

Clustering approach

A more recent progress along the direction of multi-point approach is the clustered multi-point re-weighting method [Kim03]. This approach proposed a way to use query point clusters. When user marks several points as relevant, the system clusters sets of relevant points and chooses the centroids of the clusters as their representatives, and then constructs a multi-point query using a small number of good representative points. In such a case, the weight for the query point instead of the feature can be changed based on the size of the represented cluster. The new search will be using the respective selected clusters. However, there is no mention about the search space with balanced data distribution in which clusters formation is not obvious.

To summarize major differences among the query point movement approach, the multi-point re-weighting approach, the clustering approach, and our proposed approach:

- The query point movement approach will navigate towards the target points based on the new query point after each feedback iteration, i.e. search the top k points near the new query point. But there is no search space reduction;
- The multi-point re-weighting approach will use multiple new query points instead of one new query point to navigate towards the target points, i.e. a K-NN search (find the points near to k query points). There is also no search space reduction;
- The clustering approach uses multiple new query points as well. However, instead of searching data points with the smallest total distance to all new query points, it searches the data points near to each individual representative

query point of the respective clusters separately. There is some search space reduction. But the following search may be restricted by the early clusters;

- Our proposed approach will use the multi-point re-weighting query to navigate the nearby search space of multiple query points instead of the original search space, i.e. a K-NN search inside the top k neighborhood of the multiple query points of a multi-point query. There is obviously search space reduction. At the same time, the following search will move freely according to the new query points.

In fact, we can illustrate the interactive search processes as in Figure 7.2 and Figure 7.3 to highlight the above points.

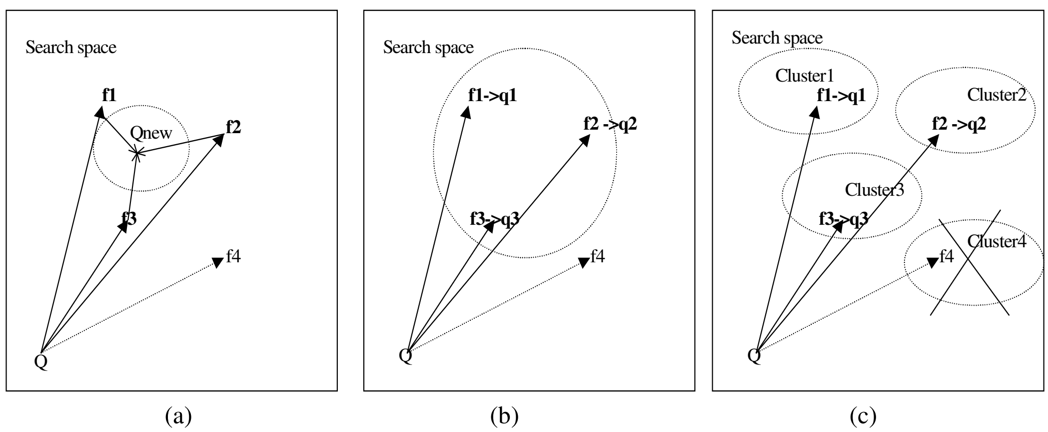


Figure 7.2. Comparison of different interactive searching processes

- In Figure 7.2(a), suppose Q is the initial query, f1, f2, f3, f4 are four outputs to the user, and f1, f2, and f3 are selected. If we use the query point movement approach, we will base on f1, f2, and f3 to generate a new query point Qnew, and find the top k points (illustrated as a small circle in Figure 7.2(a)) near to Qnew as the next round of output. It is unlikely we need to search the whole search space again to give the top k nearest neighbors of Qnew. If we employ k-regular digraph, we may use directly the data points around Qnew as the new search space. However, the problem is that Qnew may not be in the database at all since it is generated from f1, f2, and f3. So we may not use k-regular index for Qnew.

- In Figure 7.2(b), f1, f2, and f3 are still user's selections. The multi-point approach uses f1, f2, and f3 as the multi-point query, i.e. the feedbacks f1, f2, and f3 become the new query points q1, q2, and q3 respectively as shown in Figure 7.2(b). We search the neighborhood of f1, f2 and f3 at the same time to find data points that have the smallest total similarity distance to q1, q2, and q3. The search space can be very big (illustrated as a big circle in Figure 7.2(b)) if f1, f2, and f3 are not very close in the feature space.
- In Figure 7.2(c), the selections for the new query are similar to the typical multi-point approach in Figure 7.2(b), i.e. the feedbacks f1, f2, and f3 become the new query points q1, q2, and q3 respectively. However, if feature space and the distance function of the user's perception are quite different from those of the system, the initial feedbacks f1, f2, and f3 may be mapped to disjoint clusters of arbitrary shapes in the feature space as shown in Figure 7.2(c). Using the clustering approach, we eliminate clusters not involving f1, f2 and f3 from the search space and only one representative query point within each relevant cluster is considered for that cluster. In such a case, we search data points that are near to q1, or q2, or q3 separately. However, this strategy may not work well when image representation and similarity measure used for the clustering do not match perceptual characteristics exactly (since there is always a gap between the representation and perception), and the search may be trapped within the initial clusters. On the other hand, if the data points are

distributed evenly, the clusters constructed may not truly reflect the clustering, and thus the use of clusters may not be good in the search.

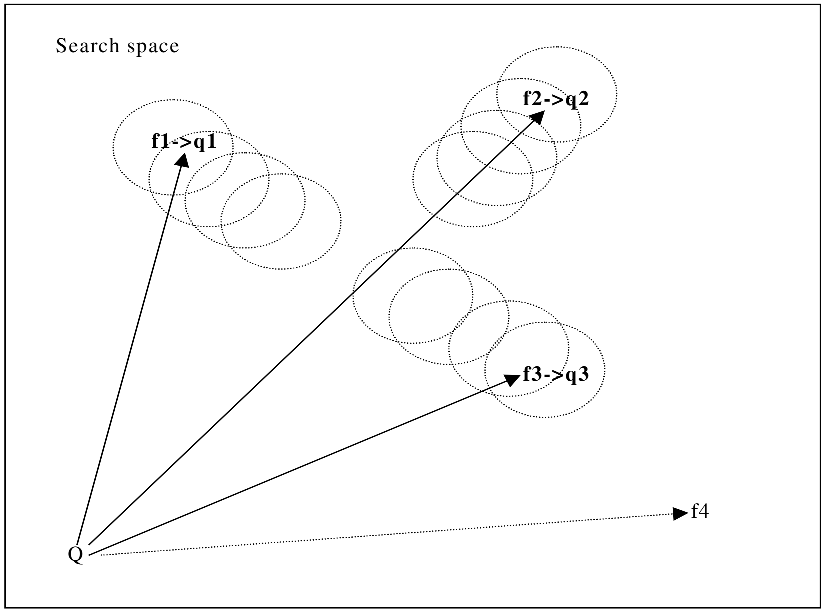


Figure 7.3. Interactive search process for proposed approach

Figure 7.3 illustrates our approach that has advantages of both the query point movement approach and the multi-point approach, and overcomes the restricted navigation region of the clustering approach. Suppose we still have f1, f2, and f3 selected and they are used as the new multi-point query. Since we have digraph indexes for f1, f2, and f3, we can search the top k nearest neighbors around f1, f2, and f3 directly. This is comparable to the clustering approach whose search space has been reduced. However, for the clustering approach, the clusters are fixed and the subsequent query points chosen will not go out of the clusters since the feedback points are required to be within the same clusters of respective query points. As for a digraph, we have individual top-k list for each data point in the search space. Therefore, even when the data do not cluster, the new query points can move towards any possible points through the reduced search space. As shown in Figure 7.3, multiple circles denote the search space for different iterations. They can move freely according to the feedback selections.

There are some other works related to search space reduction. For example, Query Refinement [Pork99]) uses auxiliary data structures such as a priority queue or a tree structure built on the fly to reduce duplicated computation. The new search space has to be computed after each feedback cycle and it is trading some percentage of recall for faster retrieval.

The recent work in [Liu06] claims to improve upon the previous approaches, and uses a Voronoi diagram to reduce the search space. The basic idea is as follow:

- A database will be sampled with k points as the initial output to user, and at the same time, a Voronoi diagram will be constructed based on the k points.
- A user inputs the feedback to choose one point (suppose PS) among the output k points.
- In the next iteration, the system will search the Voronoi cell region corresponding to PS only, i.e. use Voronoi cell region of PS as the new search space to sample k points as the next output, and repeat this process until the user find the target.

Since the re-search space is based on the Voronoi cell region, it is supposed to reduce the search space significantly after each feedback iteration. However, there are some problems in this approach comparing to our approach:

- This approach ignores the complexity of building the Voronoi diagram. The digraph approach does not need to re-build the index for subsequent retrieval cycles.

- It is not obvious on how to fit in Voronoi diagram for multi-point query and weight adjustment, i.e. how to combine the hidden information extracted from the multi-point with the Voronoi diagram.
- The approach using Voronoi diagram assumes that the user always inputs accurate single query feedback in each iteration. This is not realistic for most users. Furthermore, if the target is a similar picture not in the database, then the Voronoi diagram is not valid from the similarity point of view. It is also not efficient in searching when similar pictures are in more than one Voronoi regions.

Therefore, enhanced k-digraph plus the heuristics for the multipoint interactive similarity retrieval proposed in this thesis is a more complete solution to combine the advantages of reducing the search space and multi-point re-weighting in interactive similarity retrieval.

7.6 Evaluation and comparison experiments

We have already discussed the coverage experiment for the digraph to demonstrate the effectiveness of the interactive index approach. In the following, we use an experiment to show that the digraph indexing structure can really improve the speed for the interactive retrieval, and use a simulation to show the differences in the search space explored by different navigation approaches.

7.6.1 Response time experiment

For the ease of discussion, we use the same dataset as before for the performance experiment. In detail, the setup of the experiment is as follow:

There are 647 pictures in our database. IO&T metric is used as the base metric to build all index structures of the database. An enhanced digraph is built by adjusting the threshold to make sure that the length of each link list is less than 10. The system displays 5 candidate pictures at most each time. The interactive process is simulated systematically as follow.

Without the loss of generality, we assume the average number of feedback points selected is about 1/3 of the pictures displayed (for example, suppose 5 pictures are displayed, then 5/3 = 2 feedback points will be used). The feedback points are chosen randomly in the simulation, i.e. among the 5 outputs, the system will pick two outputs randomly as the feedback input for the next session. The average response time R_{time} to display every round of outputs will be recorded. The user input time is ignored in this case because the feedback points are generated automatically instead of manually. Note that we are not using any cache discussed in some other approaches such as [Chak04]. Therefore, there is always disk I/O instead of memory I/O when accessing the picture database. The MinMax heuristic approach and Closest Neighbor heuristic approach based on the enhanced digraph are denoted as I-MM and I-CN. The multipoint retrieval process without any index is called T-MP. The experiment is carried out using SunUltra10 and Java applet is needed. The response time are measured in seconds as shown in Figure 7.4.

From the result, we can see that without the digraph index, the retrieval time is not acceptable (Note that the first return is based on the tree structure used to organize the images, which is the same for all approaches). The MinMax heuristic approach performs slightly poorer than the Closest Neighbor heuristic approach, which is actually expected since there is almost no calculation for the latter. Since MinMax approach has the dynamic weighting feature that is more intuitive to the user and covers a bigger candidate set, the small price to pay in performance disadvantage is still acceptable.

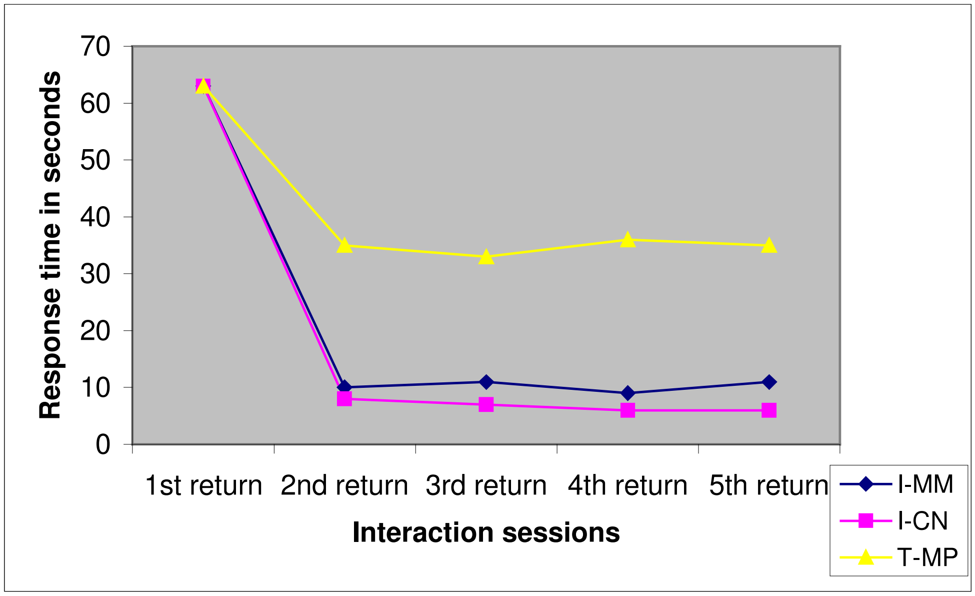


Figure 7.4. Response time (in seconds) for each displaying return

7.6.2 Experiment for search space explored

Most of the effectiveness experiments for interactive image retrieval are based on a set of target selected arbitrarily, and the rate of recall and precision are calculated based on the statistics collected through some interactions with a small group of selected users. The result obtained may have a tint of subjective factor. In our navigation experiment, we generate 2000 simulated data points randomly. This is to reduce the subjective-ness of data set selection, and to simplify the process of generating a real big iconic image database. In the experiment, we compared three navigation approaches, namely the query point movement approach, the multipoint clustering approach, and the proposed digraph approach referred as QPM, MCA, and DIG in the following. The simulated process includes the following steps:

Step 1: Mark all 2000 random data points as un-visited.

Step 2: Three data points will be randomly selected as the initial feedbacks to the similarity retrieval system.

Step 3: Based on the navigation approach selected, the system will find the similar data points using the respective measure and return the next round of candidates.

Step 4: Three un-visited data points from the return of Step 3 will be randomly chosen as the next round of feedbacks, and mark all navigated data points as visited. Go back to Step 3, and repeat the process a number of times.

We use the initial three randomly generated data points as the base to calculate the possible candidate data region CDR, the region with points that are within some similarity distance to the multiple feedback points. If a data point explored falls in CDR, we consider it as an effective visit. Otherwise, the visit is not effective (wasted). We compare QPM, MCA, and DIG in terms of the recall and the precision for each feedback session defined as follows:

$$\textbf{The recall} = \frac{\textbf{The number of effective visits} \times \textbf{100}}{\textbf{The number of data points in CDR}} \qquad \textbf{(7.5)}$$

$$\textbf{The precision} = \frac{\textbf{The number of effective visits} \times \textbf{100}}{\textbf{The number of all visits (including effective and ineffective visits)}} \qquad \textbf{(7.6)}$$

The results are compared in Figure 7.5 and Figure 7.6. It can be seen that QPM has the best precision in the early interaction sessions. Since the query point can move out of CDR, the precision drops a lot in the later sessions. However, the real problem with QPM is how to reduce the search space. MCA can't improve its recall after first few iterations as its search space has been restricted. Thinking comparison, DIG has the benefits of integrating the advantages of both QPM and MCA approaches in terms of usability and flexibility for interactive similarity navigation.

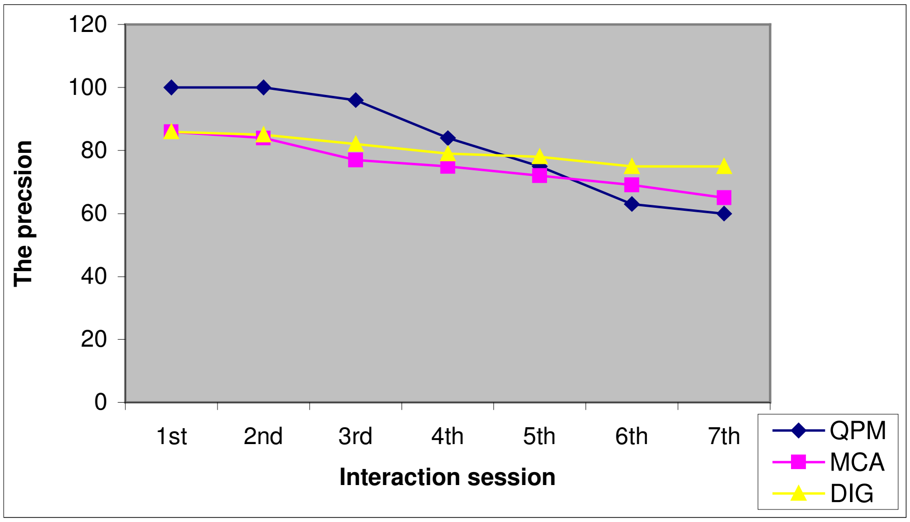


Figure 7.5. The precision comparison for each interaction session

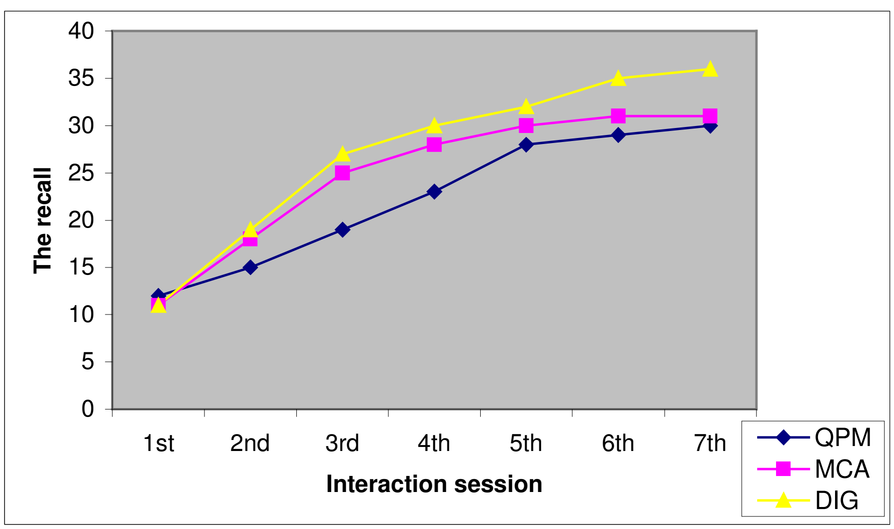


Figure 7.6. The recall comparison for each interaction session

Chapter 8

Conclusion and future work

In the last chapter, we will conclude our research starting from the general iconic picture representations and measures for automatic similarity retrieval to the interactive fine-tune process for subjective navigation in result set, and address some possible future work that can be explored.

In this thesis, we analyzed 2D space spatial relation representation and similarity retrieval. We introduced some qualitative spatial relation representations that include using two 1D spatial relations to represent 2D spatial relations, and using systematic topological and orientation relations to represent 2D space spatial relation. We also proposed an augmented spatial relation representation to include the extent of objects to complement the missing information in the topological and orientation representation. We compare the pros and cons of different proposals.

Based on each of our discussed spatial relation representations, we introduced the respective similarity measure and described the retrieval algorithms needed. Some experiments are carried out and a prototype is built to demonstrate the feasibility of turning the discussed representation and retrieval approaches into a full fledge picture similarity retrieval system.

To make the similarity retrieval system more applicable, human decision is added to the retrieval procedure in the later discussion. Instead of completely decided by the system on what to retrieve, users are involved through their feedback. While an automatic retrieval system provides some first level filtering functionality that can be adapted to some general fuzzy retrieval, user feedback can make the system more accurate and intuitive for some personalized retrieval.

Finally, we introduced some advanced topics for the interactive similarity retrieval process. The multipoint query concept is discussed as a base for some dynamic and adjustable measure, and the use of a dynamic measure will enable a retrieval system to react more intelligently upon user's feedback. We summarize our contributions and highlight some interesting topics we are still exploring as follow.

8.1 Main contributions

In summary, we have proposed in this thesis:

- The scheme of IO&T representation to represent the spatial knowledge qualitatively

The orientation spatial relationships between two objects are defined using two intrinsic orientation relations. This representation captures not only the normal spatial relations among objects but also the rotation of objects. This approach overcomes the ambiguity problems that exist in some other representations, and most importantly, it is invariant with respect to translation and rotation of the picture.

- IO&T similarity measure

This measure captures not only how many objects have changed their spatial relations with respect to dominant objects, but also the degree of these changes, which makes the measure of picture similarity more meaningful and more accurate.

- A simulated similarity retrieval algorithm

This algorithm is a more practical solution of similarity retrieval as compared to the use of the common clique algorithm whose complexity is NP-hard. It is especially useful for similarity retrievals involving only a small number of displaced objects.

- An object based similarity retrieval approach

This is a concept to make use of dominant objects in the search of similar pictures. When we have a different sequence of dominant objects, the list of similar pictures output by the algorithm will change accordingly. The algorithm trims the answer set of similar pictures very effectively with the use of IO&T similarity measure. It is useful for similarity retrievals involving only a small number of dominant objects.

- Augmented spatial relation representation

Comparing to existing systems, the augmented spatial relation representation approach is not only rotation invariant, but also captures the relative distance and orientation range between objects. This approach overcomes the ambiguity problems that exist in other orientation representations, and is more flexible and applicable.

- Regular digraph indexing of a pictorial database for interactive similarity retrieval

The proposed interactive similarity retrieval approach makes use of the feedback from the user, and avoids the high cost of re-computation of similarity distances. The interactive similarity retrieval process is similar to the navigation auto-piloted by the system measure and guided by the user in the image database. The retrieval algorithm proposed prevents looping and guarantees to find the target image. In particular, the index structure and the respective retrieval approach are straightforward and can be adapted to different similarity measure.

- Dynamic similarity metrics for the multipoint feedback based similarity retrieval

Multipoint feedback concept opens a door for dynamic similarity measure. The use of similarity information common to the multiple points query makes the interactive retrieval process more intelligent. The implicit knowledge from the user will be re-used for the interactive retrieval without the loss of the objectiveness for the system measure. Combining with the enhanced digraph index structure, the dynamic multi-point retrieval takes the advantage of both search space reduction and effective query point movement.

- Self-adjustable indexing structure through learning from the feedbacks

Typically, a learning system tends to be subjective though it provides extra value for a user some time. What we have introduced in this thesis is a general system for multi-users instead of for a single user. The proposed adjustment approach fine-tune the index by adding only some weight to the more popular ones among those tied items according to the system measures. There is no loss of objectiveness from the original system in this learning process.

- Image similarity retrieval system prototype

Finally, we have implemented a prototype to demonstrate the feasibility of integrating the concepts, structures and algorithms proposed into a workable similarity retrieval system.

8.2 Future research issues

We have discussed many topics related to picture representation and similarity retrieval and proposed some solutions. However, there are some outstanding issues not covered and will be studied in future:

3D space representation

There are already some works regarding 3D space spatial relation representation such as [Schw91] and [Delb93] etc. With the use of the intrinsic orientation, the IO&T representation can be extended to 3D space. Since the application area of 3D space representation is quite wide, the extension of IO&T representation to 3D space can be very interesting.

Same objects appear in one picture

In this thesis, we have not discussed how to deal with multiple occurrences of an object in one picture. This situation will normally cause ambiguity problem. There are some existing discussions (such as [Chan81] and [Cost92]) to solve this problem in other representations. Some of these methods may be revised and adapted to the discussed representations.

Spatial relations reasoning by constraints

Constraint reasoning is a very useful in real life and it has become a hot topic recently. In spatial relations reasoning, constraint reasoning is to find some spatial relations by using some given spatial relations as constraints. For example, suppose we know two constraints: *Object A is on the left of Object B, and Object B is on the left of Object C*. Then Object A can only be on the left of C in order to satisfy both constraints. Some researchers have done some works in this area such as [Shar95] and [Sist95] etc. However, more efforts are still required to build a more practical system.

Learning from feedback

How to make a system learn is always an interesting and yet very hard problem to solve. For example, how to make the system adapt itself in selecting the right path through the index structure to move towards the target images, what are the heuristics that can be used to guide the system to learn and adjust the index based on the users' feedback etc. This is a topic involving AI and psychology which has attracted a lot of interests. Although we have some discussions in Chapter 7, there are still much more to be explored in future.

References

[Agga01] Aggarwal, C.C., "A human-computer cooperative system for effective high dimensional clustering", *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, San Francisco, California, United States, pp.221-226, 2001

[Alle83] Allen, J.F., "Maintaining knowledge about temporal intervals", *Communications of the ACM*, Vol. 26, No. 11, pp.832-843, 1983

[Ang98] Ang, C.H., T.W. Ling, and X.M. Zhou, "Qualitative spatial relationships representation IO&T and its retrieval", *9th International Conference of Database and Expert Systems Applications*, Vienna, Austria, 1998, Lecture Notes in Computer Science 1460, pp.270-279

[Anna00] Annamalai, M., R. Chopra, S. DeFazio, and S. Mavris, "Indexing images in Oracle8i", *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, United States, pp.539-547, 2000

[Bach96] Bach, J.R., C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain, and C. Shu, "Virage image search engine: An open framework for image management", *Proceedings of SPIE – Storage & Retrieval for Still Image and Video Database IV*, pp.76-87, 1996

[Bart01] Bartolini, I., P. Ciaccia, and F. Wass, "FeedbackBypass: A new approach to interactive similarity query processing", *Proceedings Of the 27th VLDB Conference*, Roma, Italy, pp.201-210, 2001

[Bhan94] Bhanu, B and S. Lee, *Genetic learning for adaptive picture segmentation*, Kluwer Academic, Norwell, 1994

[Bron73] Bron, C., J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph", *Communications of the ACM*, Vol. 16, No. 9, pp575-577, 1973

[Buck95] Buckley, C. and G. Salton, "Optimization of relevance feedback weights", *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, United States, pp.351-357, 1995

[Chak04] Chakrabarti, K., M. Oretega-Binderberger, S. Mehrotra, and K. Porkaew, "Evaluating refined queries in top-k retrieval systems", *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No. 2, pp.256-270, 2004

[Chan90] Chandrasekaran, B. and N.H. Narayanan, "Towards a theory of common sense visual reasoning", *Foundations of Software Technology and Theoretical Computer Science, 10th Conference,* Bangalore, India, pp.388-409, 1990

[Chan81] Chang, S.K. and T. Kunii, "Pictorial database systems", *Computer (Special Issue on Pictorial Information Systems)*, S.K. Chang, Ed., pp.13-21, November, 1981

[Chan87] Chang, S.K., Q.Y. Shi, and C.W. Yan, "Iconic indexing by 2D strings", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 9., No. 3, pp.413-428, 1987

[Chan88] Chang, S.K. and Y. Li, "Representation of multi-resolution symbolic and binary pictures using 2D-H strings", *Proceedings of IEEE workshop on languages for Automata,* Maryland, pp.190-195, 1988

[Chan91a] Chang, C.C., "Spatial match retrieval of symbolic pictures", *Journal of Information Science and Engineering*, Vol. 7, pp.405-422, 1991

[Chan91b] Chang, C.C. and S.Y. Lee, "Retrieval of similar pictures on pictorial databases", *Pattern Recognition*, Vol. 24, No. 7, pp.675-680, 1991

[Chan92] Chang, C.C., "A fast algorithm to retrieve symbolic pictures", *International Journal of Computer Mathematics,* Vol. 43, No. 1, pp.133-138, 1992

[Chan96] Chang, C.C. and D.C. Lin, "A spatial data representation; an adaptive 2D-H string", *Pattern Recognition Letters*, Vol. 17, Issue 2, pp.175-185, Feb. 1996

[Chan00] Chang, E., B. Li and C. Li, "Towards perception-based picture retrieval", *IEEE Content-Based Access of Picture and Video Libraries*, pp.010-105, June 2000

[Ciac97] Ciaccia, P., M. Patella and P. Zezula, "M-tree: an efficient access method for similarity search in metric spaces", *Proceedings of the 23rd VLDB Conference*, Athens, Greece, pp.426-435, 1997

[Cich80] Cichelli, R., "Minimal perfect hash functions made simple", *Communications of the ACM,* Vol. 23, No. 1, pp.17-19, 1980

[Cook82] Cook, C.R. and R. Oldehoeft, "A letter-oriented minimal perfect hashing function", *ACM SIGPlan Notices,* Vol. 17, No. 9, pp.18-27, 1982

[Cost92] Costagliola, G., G. Tortora and T. Arndt, "A unifying approach to iconic indexing for 2-D and 3-D scenes", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4. No. 3, pp.205-222, 1992

[Delb93] Del Bimbo, A., M. Campanai, and P. Nesi, "A three-dimension iconic environment for image database querying", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19. No. 10, pp.997-1011, 1993

[Dilw02] Dilworth, J., "Varieties of Visual Representation", *Canadian Journal of Philosophy 32,* No. 2, pp.183-205, June 2002

[Disc02] Di Sciascio, E., F.M. Donini, and M. Mongiello, “Spatial layout representation for query-by-sketch content-based picture retrieval”, *Pattern Recognition Letters*, Vol. 23, Issue 13, pp.1599-1612, Nov. 2002

[Egen91] Egenhofer, M.J., “Point-set topological spatial relations”, *International Journal of Geographical Information Systems*, Vol. 5., No. 2, pp.161-174, 1991

[Elkw99] El-Kwae, E.A., M.R. Kabuka, “A robust framework for content-based retrieval by spatial similarity in picture database”, *ACM Transactions on Information Systems*, Vol. 17, No. 2, pp.174-198, 1999

[Elkw00] El-Kwae, E.A., M.R. Kabuka, “Efficient content-based indexing of large picture databases”, *ACM Transactions on Information Systems*, Vol. 18, No. 2, pp.171-210, 2000

[Flic95] Flickner, M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafnewr, D. Lee, D. Petkovic, D. Streele, and P. Yanker, “Query by image and video content: The QBIC system”, *Computer*, Vol. 27, No. 9, pp.23-32, 1995

[Forb91] Forbus, K.D., P. Nielsen, and Falting, B., “Qualitative spatial reasoning: The clock project”, *Artificial Intelligence,* Vol. 51, pp.417-471, 1991

[Gudi95] Gudivada, V., and V. Raghavan, “Design and evaluation of algorithm for image retrieval by spatial similarity”, *ACM Transactions on Information Systems*, Vol. 13, No. 1, pp.115-144, 1995

[Gudi98] Gudivada, V., "θℛ-String: A geometry-based representation for efficient and effiective retrieval of image by spatial similarity", *IEEE Transactions on Knowledge and Data Engineering*, Vol.10, No. 3, pp.504-512, 1998

[Gues89] Guesgen, H.W., "Spatial reasoning based on Allen's temporal logic", *Technical Report TR-89-049*, ICSI, Berkeley, CA, 1989

[Guti88] Guting, R, "Geo-relational algebra: A model and query language for geometric database systems", *Advances in database technology, EDBT' 88, International Conference on Extending Database Technology*, pp.506-527, Berlin, 1988

[Gutt84] Guttman, A., "R-Trees: A dynamic index structure for spatial searching", *Proceeding of the 1984 ACM-SIGMOD, International Conference of Management of Data,* Boston, Massachusetts, pp.47-57, 1984

[Hern94] Hernandez, D., *Qualitative representation of spatial knowledge*, Lecture Notes in Artificial Intelligence, No. 804, Springer -Verlag, 1994

[Hogg91] Hogg, S and I. Schwarzer, "Composition of spatial relations", *Forschungsberichte Kunstliche Intelligenz* FKI-163-91, Institut fur Informatik, Technishe Universitat Munchen, (44), 1991

[Hua06] Hua, K.A., N. Yu, and D. Liu, "Query decomposition: a multiple neighborhood approach to relevance feedback processing in content-based image retrieval", *Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006),* Atlanta, GA, 2006

[Huan94] Huang, P.W. and Y. R. Jean, "Using 2D C+-strings as spatial knowledge representation for picture database system", *Pattern Recognition,* Vol. 27, No. 9, pp.1247-1257, 1994

[Huan95] Huang, P.W. and Y. R. Jean, "Reasoning about pictures and similarity retrieval for picture information systems based on SK-set knowledge representation", *Pattern Recognition,* Vol. 28, No. 12, pp.1915-1925, 1995

[Huan96] Huang. P.W. and Y.R. Jean, "Spatial reasoning and similarity retrieval for picture database systems based on RS-string", *Pattern Recognition*, Vol. 29, No. 12, pp.2103-2114, 1996

[Huan03] Huang. P.W. and S.K. Dai, "Image retrieval by texture similarity", *Pattern Recognition*, Vol. 36, pp.665-679, 2003

[Huan04] Huang. P.W. and C.H. Lee, "Image database design based on 9D-SPA representation for spatial relations", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 12, pp.1486-1496, 2004

[Ishi98] Ishikawa, Y., Subramanya, R., and Faloutsos, C., "Mindreader: Querying databases through multiple examples", *Proceedings of the 24th VLDB Conference,* New York City, New York, USA, pp.218-227, 1998

[Kim03] Kim, D.H., and C.W., Chung, "Qcluster: Relevance feedback using adaptive clustering for content-based picture retrieval", *Proceedings of the 2003 ACM SIGMOD,* San Diego, CA, pp.599-611, 2003

[Lee90] Lee, S.Y. and F.J. Hsu, "2D C-String: A new spatial knowledge representation for picture database systems", *Pattern Recognition*, Vol. 23, No.10, pp.1077-1087, 1990

[Lee91] Lee, S.Y. and F.J. Hsu, "Picture algebra for spatial reasoning of iconic pictures represented in 2D C-string", *Pattern Recognition Letters,* Vol. 12, Issue 7, pp.425-435, 1991

[Lee92] Lee, S.Y. and F.J. Hsu, "Spatial reasoning and similarity retrieval of pictures using 2D C-string knowledge representation", *Pattern Recognition,* Vol. 25, No. 3, pp.305-318, 1992

[Lee04] Lee, J.Y., S.Y. Jeong, K.S. Han, and B.T. Chun, "Image navigation: a massively interactive model for similarity retrieval of images", *International Journal of Computer Vision,* Vol. 56, No. ½, pp.131-145, 2004

[Li02] Li, C, E. Chang, H. Garcia-Molina, J.Z. Wang, and G. Wiederhold, "Clustering for approximate similarity search in high-dimensional spaces", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 4, pp.792-808, 2002

[Lim01] Lim, J.H., J.K. Wu, S. Singh, and D. Narasimhalu, "Learning similarity matching in multimedia content-based retrieval", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 5, pp.846-850, 2001

[Liu06] Liu, D., K.A. Hua, K. Vu, and N. Yu, "Efficient target search with relevance feedback for large CBIR systems", *The 21st Annual ACM Symposium on Applied Computing (SAC 2006)*, Dijon, France, pp.1393-1397, April 2006

[Lu00] Lu, Y., C.H. Hu, X.Q. Zhu, H.J. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems", *Proceedings of the eighth ACM international conference on Multimedia*, Marina del Rey, California, United States, pp.31-37, 2000

[Mao03] Mao, R., W. Xu, N. Singh and D. P. Miranker, "An assessment of a metric space database index to support sequence homology", *Proceedings of 3rd IEE Symposium on BioInformatics and BioEngineering*, Washington, DC, USA, pp.375-384, 2003

[Nabi95] Nabil, M., J. Sheperd and A. H.H. Ngu, "2D projection interval relationships: A symbolic representation of spatial relationships", *Advances in Spatial Databases: Fourth International Symposium, SSD'95*, Lecture Notes in Computer Science, No.951, pp.292-309, Springer-Verlag, 1995

[Nabi96] Nabil, M., A. H.H. Ngu and J. Sheperd, "Picture similarity retrieval using 2D projection interval representation", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 4, pp.533-539, 1996

[Nabi97] Nabil, M., "Modelling and retrieving multimedia data using spatial and temporal information", *Ph.D thesis*, School of Computer Science and Engineering, University of New South Wales, 1997

[Papa94] Papadias, D. and T. Sellis, "A pictorial language for the retrieval of spatial relations from the picture databases", *Proceedings Of the 6th International Symposium on Spatial Data Handling*, Taylor Francis, Edinburgh, Scotland, pp.476-488, 1994

[Papa98] Papadias, D., N. Mamoulis and V. Delis, "Algorithms for querying by spatial structure", *Proceedings of the 24th VLDB Conference,* New York City, New York, USA, pp.546-557, 1998

[Petr93] Petraglia, G., M. Sebillo, M. Tucci and G. Tortora, "Towards normalized iconic indexing", *Proceedings of the 1993 IEEE Symposium on Visual Language*, Bergen, Norway, pp.392-394, 1993

[Petr94] Petraglia, G., M. Sebillo, M. Tucci and G. Tortora, "Rotation invariant iconic indexing for picture database retrieval", *Progress in Picture Analysis and Processing III* (edited by S. Impedovo), World Scientific, pp.271-278, 1994

[Peuq87] Peuquet, D.J. and Z. Ci-Xiang, "An algorithm to determine the directional relationship between arbitrary-shaped polygons in the plane", *Pattern Recognition*, Vol. 20. No. 1, pp.65-74, 1987

[Pork99] Porkaew, K., Chakrabarti, K., and Mehrotra, S., "Query refinement for multimedia similarity in MARS", *ACM International Multimedia Conference*, Orlando, Florda, USA, pp.235-238, 1999

[Rocc71] Rocchio, J.J. Jr., "Relevance feedback in information retrieval", *in G. Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing*, pp.313-323. Prentice Hall, 1971

[Rui98] Rui, Y., T.S., Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based picture retrieval", *IEEE Transactions on Circuits and Video Technology*, 8(5), pp.644-655, September 1998

[Same84] Samet, H., "The quadtree and related data structures", *ACM Computing Survey,* Vol. 16, No. 2, pp.187-260, 1984

[Schw91] Schwarzer, I. and S. Hogg, "Komposition raumlicher Relationen", *Fortgeschrittenenpraktikum*, Institut fur Informatik, Technische Universitat Munchen, 1991

[Shar95] Sharma, J. and D. M. Flewelling, "Inferences from combined knowledge about topology and directions", *Advances in Spatial Databases: Fourth International Symposium, SSD'95*, Lecture Notes in Computer Science No.951, pp.279-291, Springer-Verlag, 1995.

[Sist95] Sistla, A.P., C. Yu, C. Liu and K. Liu, "Similarity based retrieval of pictures using indices on spatial relationships*", Proceedings of the 21st VLDB Conference*, Zurich, Switzerland, pp.619-629, 1995

[Smit96] Smith, J.R. and S.F. Chang, "VisualSEEK: A full automated content-based image query system", *Proceedings of the 4th ACM International Multimedia Conference*, pp.87-98, 1996

[Smyt03] Smyth, W.F., C.P. Lam, X. Chen, and V. Maxville, "Heuristics of image retrieval using spatial configurations", *Proceedings of the VIIth Digital Image Computing: Techniques and Applications*, Sydney, Australia, pp.909-918, 2003

[Tani76] Tanimoto, S.L., "An iconic symbolic data structuring scheme", *Pattern Recognition and Artificial Intelligence*, Academic Press, New York, pp.452-471, 1976

[Tani98] Taniguchi, N. and M. Yamamuro, "Multiple inverted array structure form similar image retrieval", *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp.160-169, Austin, TX, USA, 1998

[Tucc91] Tucci, M., G. Costagliola and S. K. Chang, "A remark on NP-completeness of picture matching", *Information Processing Letters,* Vol. 39, No. 5, pp.241-243, 1991

[Wu94] Wu, T.C. and C.C. Chang, "Application of geometric hashing to iconic database retrieval", *Pattern Recognition Letters,* Vol. 15, Issue 9, pp.871-876, 1994

[Wu00] Wu, L.J., C. Faloutsos, K. Sycara, T.R. Payne, "FALCON: Feedback adaptive loop for content-based retrieval", *Proceedings of the 26st VLDB Conference*, Cairo, Egypt, pp.297-306, 2000

[Wu01] Wu, M.F., M. Fuller, and R. Wilkinson, "Using clustering and classification approaches in interactive retrieval", *Information Processing and Management*, Vol. 37, pp.459-484, 2001

[Zhan03] Zhang, Q.L. and S.S.-T. Yau, "A general approach to indexing and retrieval of images in image databases", *Communications in Information and Systems*, Vol. 3, No. 1, pp.61-74, June 2003

[Zhou97] Zhou, X.M. and C.H. Ang, "Retrieving similar pictures from a pictorial database by an improved hashing table", *Pattern Recognition Letters,* Vol.18, Issue 8, pp.751-758, 1997

[Zhou00] Zhou, X.M., C.H. Ang, T.W. Ling, "Applying Augmented Orientation Spatial Similarity Retrieval in Pictorial Database", *Proceedings of the Fifth Working Conference on Visual Database Systems (VDB5)*, Fukuoka, Japan, pp.163-176, May 2000

[Zhou01] Zhou, X.M., C.H. Ang, and T.W. Ling, "Image retrieval based on object's orientation spatial relationship", *Pattern Recognition Letters,* Vol. 22, Issue 5, pp.469-477, 2001

[Zhou02] Zhou, X.M., C.H. Ang, and T.W. Ling, "Towards Interactive Spatial Relationship Retrieval", *The 6th World Multi-conference on Systemics, Cybernetics, and Informatics (SCI 2002)*, Sheraton World Resort, Orlando, Florida, USA, July 15, 2002

[Zhou04] Zhou, X.M., C.H. Ang, and T.W. Ling, "Indexing iconic image database for interactive spatial similarity retrieval", *the 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004),* Jeju Island, Korea, pp.314-324, March 2004

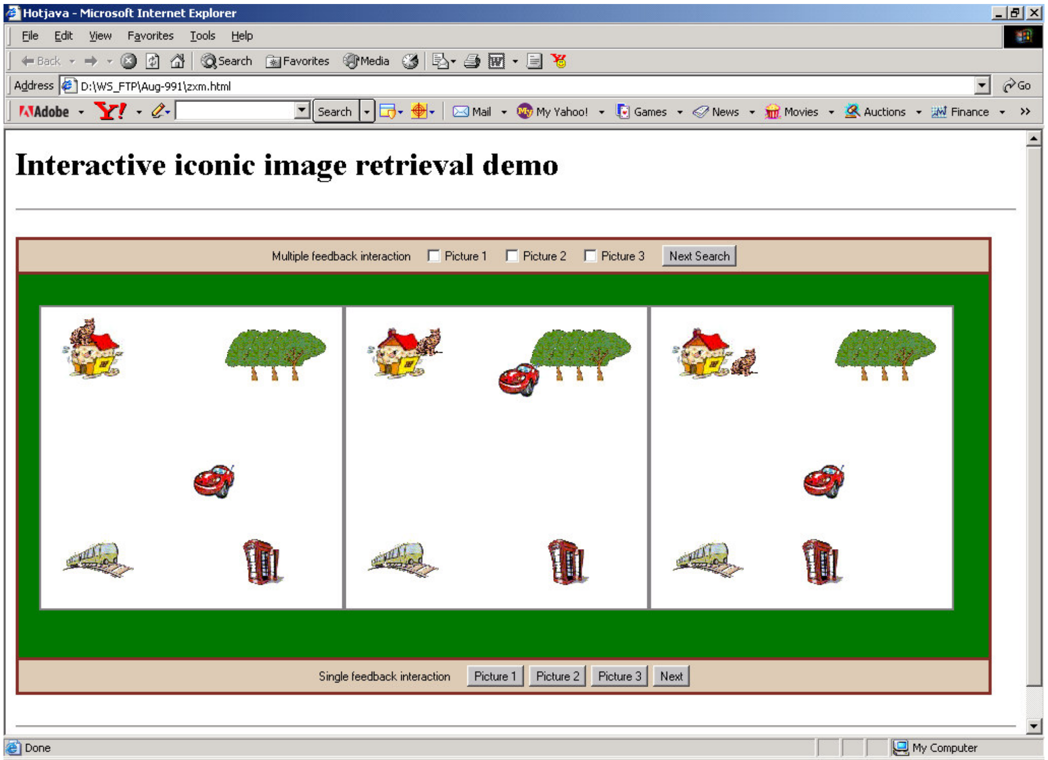
[Zhou06] Zhou, X.M., C.H. Ang, and T.W. Ling, "Dynamic interactive spatial similarity retrieval in iconic image database using enhanced digraph", *The 21st Annual ACM Symposium on Applied Computing (SAC 2006)*, Dijon, France, pp.46-51, April 2006.

Appendix

We include some screen shots for the interactive iconic similarity retrieval demo in the following. The demo is to illustrate some interactive processes discussed in this thesis. IO&T metrics is used as the measure at this stage. A more sophisticated system can be built based on the prototype though. We use pictures that allow us to identify the difference quickly. Only six objects are used in the demo and three pictures are displayed each time to make the display clearer for the demonstration (Note that the icon for three trees are considered as one aggregated object instead of multiple objects). We use Java applet to implement the simple interface that is similar to some internet image search engines such as PicSearch at http://www.picsearch.com/. In the example, we use object-based retrieval in which the house is the dominant object.

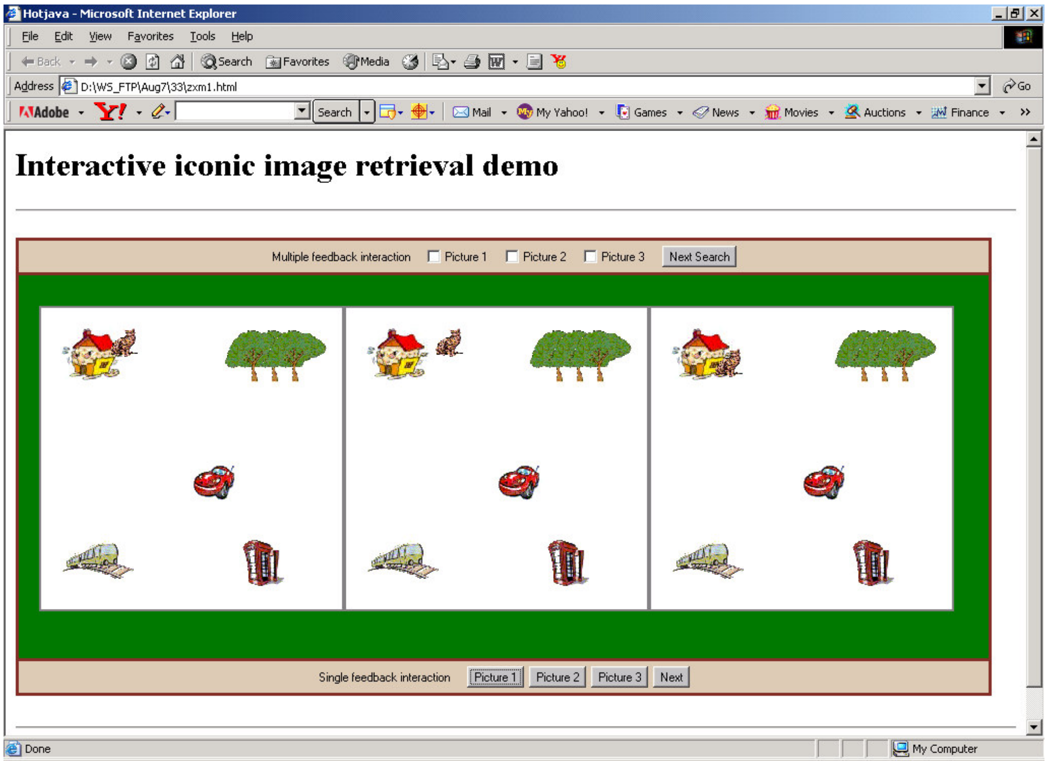
The first screen shot is supposed to be the returns from the automatic retrieval. There are three candidate pictures displayed. The user has options to use either the buttons in the lower panel to input "single feedback" or the check buttons in the upper panel to select "multiple feedbacks".

Screen Shot 1:



We consider the "single feedback" first in this case. There are four buttons in the lower panel. One button among "Picture 1", "Picture 2", or "Picture 3" can be clicked if the respective candidate picture looks more similar to the wanted picture based on the user's perception. If there is no "good" return among these three, the user can select "next" button to display the next three candidates. In this case, we suppose the user click the "Picture 1" button in the lower panel. The system uses the stored digraph index linked from "Picture 1" to return the next three candidate pictures quickly as shown in Screen Shot 2.

Screen Shot 2:

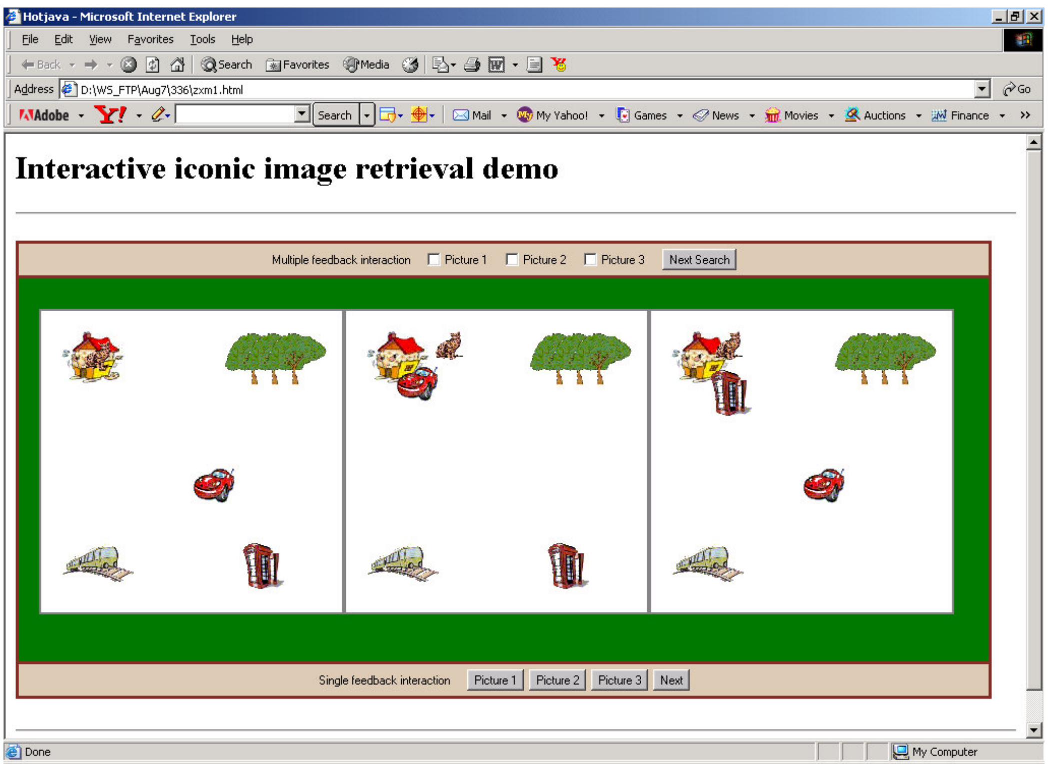


All three returns look similar to the previous "Picture 1". Since the house is the dominant object, we have some orientation relation changes between the House and the Cat for the new "Picture 1" and "Picture 3" with regard to the old "Picture 1". At the same time, there are both topological relation and orientation relation change between the House and the Cat for the new "Picture 2" with regard to the old "Picture 1". At this stage, if the user uses "multiple feedbacks", then there are few options for the user to choose.

- If no candidate picture is chosen or all candidate pictures are chosen, the system will just display next three candidate pictures after the "Next Search" button is clicked as no additional information can be extracted from the feedback. This is not a good case.

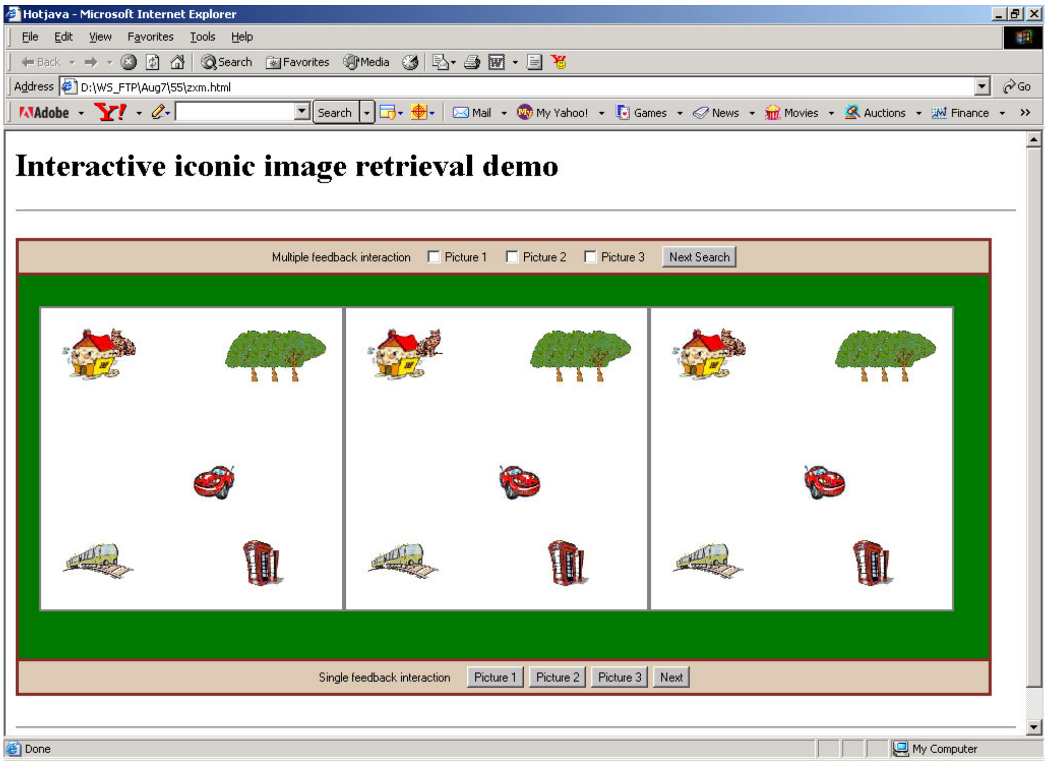
- If only one picture is selected, the system will go back to "single feedback" process. Since we are using Checkbox, the user is supposed to click "Next Search" button.
- The interesting part is that two of the three pictures are selected when the user click the "Next Search" button.

Screen Shot 3:



Screen Shot 3 is the result by selecting "Picture 1" and "Picture 2" in Screen Shot 2. Comparing the previous "Picture 1" and "Picture 2" in Screen Shot 2, there is no orientation relation change but only topological relation change between the House and the Cat. Based on the heuristics discussed, the system will adjust the weight for the topological similarity and the orientation similarity respectively. Obviously, the new returned candidate pictures have more topological relation changes instead of orientation relation changes between the House and other objects. If this is not the case, then perhaps the user select "Picture 1" and "Picture 3" in Screen Shot 2.

Screen Shot 4:



Screen Shot 4 is the result by selecting "Picture 1" and "Picture 3" in Screen Shot 2. Comparing the "Picture 1" and "Picture 2" in Screen Shot 2, there is no topological relation change but only orientation relation change between the House and the Cat. Based on the heuristics discussed, the system will increase the weight for the topological similarity and decrease the weight for the orientation similarity respectively. In such a case, the new returned candidate pictures look all similar except that some objects changed their direction (such as the Car and the Cat).

From the demo, we can see that the interactive retrieval is a subjective process. Although the demo is for the illustration purpose, the real system can be more interesting with the combination of different similarity measure on different dimension. Note that the system similarity measure is very important though the heuristics are used to fine-tune the similarity metrics during the retrieval.