Chapter 2

Symbolic Projections

2.1 Introduction

What are symbolic projections? How can symbolic projections be applied to pictorial information retrieval and spatial reasoning? A simple example will first be presented to illustrate the concept.

Fig. 2.1(a) shows a picture with a house, a car and a tree. This picture is called a *symbolic picture*, as opposed to an actual image, because it contains objects that have symbolic names: house, tree, car, etc. Suppose the objective is to find out whether there is a tree to the southeast of the house. The x-projection of the above symbolic picture can be constructed as follows:

The names of objects in each column of the symbolic picture are projected onto the x-axis. The $<$ symbol is inserted to distinguish the objects belonging to different columns. Thus the x-projection is:

x-projection: house car $<$ tree

Similarly, the y-projection is:

y-projection: house $<$ tree $<$ car

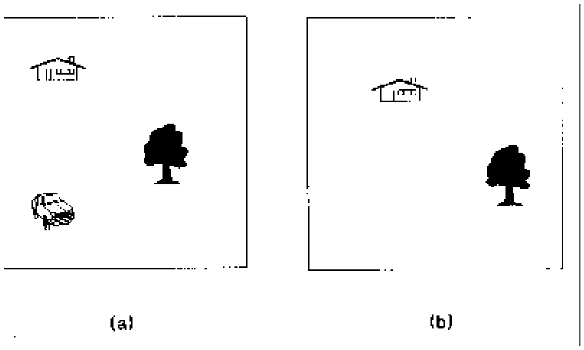


FIGURE 2.1. A symbolic picture (a) and its subpicture (b).

Unlike the projections of a mathematical function, the projections of a symbolic picture are strings. A pair of two symbolic projections is called a *2D string* [2].

The statement “there is a tree to the southeast of a house” corresponds to the symbolic picture shown in Fig. 2.1(b). This picture has the following symbolic projections:

x-projection: house $<$ tree

y-projection: house $<$ tree

We immediately notice “house $<$ tree” is a subsequence of “house car $<$ tree” and “house $<$ tree” is a subsequence of “house $<$ tree $<$ car”. In this case, the two symbolic pictures can be perfectly reconstructed from the two corresponding pairs of symbolic projections. Therefore, the above statement can be verified to be true, just by checking the subsequence property of the 2D strings involved.

The Theory of Symbolic Projection was first developed by Chang and co-workers [2] based upon the above described intuitive concept. It forms the basis of a wide range of image information retrieval algorithms. It also supports pictorial-query-by-picture, so that the user of an image information system can simply draw a picture and use the picture as a query.

Many researchers have since extended this original concept, so that there is now a rich body of theory as well as empirical results. The extended Theory of Symbolic Projection can deal with not only point-like objects, but also objects of any shape and size [3, 4]. Moreover, the Theory can deal with not only one symbolic picture, but also multiple symbolic pictures, three-dimensional pictures, a time sequence of pictures, etc. [5].

The purpose of this chapter is to introduce the elements of Symbolic Projection Theory as a complement of the Theory of Discrete Computed Tomography. 2D string representations of symbolic pictures are described in Section 2.2 . The matching of pictures using 2D strings is described in Section 2.3. The remaining five sections review the various applications of symbolic projections.

2.2 2D string representations of symbolic pictures

Let Σ be a set of symbols, or the vocabulary. Each symbol could represent a pictorial object, a pixel, etc.

Let A be the set { ’=’ , ’<’, ’:’ }, where ’=’, ’<’ and ’:’ are three special symbols not in Σ. These symbols will be used to specify spatial relationships between pictorial objects.

A *1D string* over Σ is any string $x_1\ x_2\ \ldots x_n$, $n \geq 0$, where the x_i’s are in Σ.

A *2D string* over Σ, written as (u,v), is defined to be
$(x_1\ y_1\ x_2\ y_2\ \ldots y_{n-1}\ x_n,\ x_{p(1)}\ z_1\ x_{p(2)}\ z_2\ \ldots .\ z_{n-1}\ x_{p(n)})$

where

$x_1 \ldots x_n$ is a 1D string over Σ;

p: $\{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ is a permutation over $\{1,\ldots, n\}$;

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

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

We can use 2D strings to represent pictures in a natural way. As an example, consider the picture shown in Fig. 2.2.

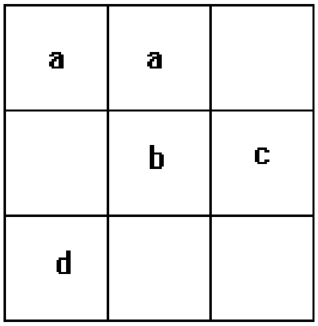


FIGURE 2.2. A picture f.

The vocabulary is $\Sigma = \{a, b, c, d\}$. The 2D string representing the above picture f is,
$(a = d < a = b < c \ , \ a = a < b = c < d)$
$= (x_1 \ y_1 \ x_2 \ y_2 \ x_3 \ y_3 \ x_4 \ y_4 \ x_5, \ x_{p(1)} \ z_1 \ x_{p(2)} \ z_2 \ x_{p(3)} \ z_3 \ x_{p(4)} \ z_4 \ x_{p(5)})$
where
$x_1 \ x_2 \ x_3 \ x_4 \ x_5$ is $adabc$;
$p(1)=1$, $p(2)=3$, $p(3)=4$, $p(4)=5$, $p(5)=2$;
$x_{p(1)} \ x_{p(2)} \ x_{p(3)} \ x_{p(4)} \ x_{p(5)}$ is $aabcd$;
$y_1 \ y_2 \ y_3 \ y_4$ is $= < = <$;
$z_1 \ z_2 \ z_3 \ z_4$ is $= < = <$.

In the above, the symbol ’<’ denotes the left-right spatial relation in string u, and the below-above spatial relation in string v. The symbol ’=’ denotes the spatial relation “approximately at the same spatial location as”. The symbol ’:’ denotes the relation ”in the same set as”. Therefore, the 2D string representation can be seen to be the *symbolic projection* of picture f along the x- and y- directions.

In the 2D string representation, the operators ’=’ can be omitted. Therefore in the above example, the 2D string can be rewritten as $(ad < ab < c$,

$aa < bc < d$).

If we are only interested in the *relative spatial relationships* between objects, we can rewrite ’<<’ to ’<’ to obtain the *reduced 2D string.* For example, the reduced 2D string of ($a << b$,ab) is ($a < b$,ab). Other types of 2D strings can be found in [2].

A *symbolic picture* f is a mapping M x $M \rightarrow W$, where $M = \{1, 2, \ldots,$ m}, and W is the power set of Σ (the set of all subsets of V). The empty set ϕ then denotes a null object. In Fig. 2.2, the “blank slots” can be filled by empty set symbols, or null objects. The above picture is,

$f(1,1) = \{\text{a}\}$ $f(1,2) = \phi$ $f(1,3) = \{\text{d}\}$;
$f(2,1) = \{\text{a}\}$ $f(2,2) = \{\text{b}\}$ $f(2,3) = \phi$;
$f(3,1) = \phi$ $f(3,2) = \{\text{c}\}$ $f(3,3) = \phi$.

It is easy to see that from f, we can construct the 2D string (u,v). The above example already illustrates the algorithm. Conversely, from the 2D string (u,v), we can reconstruct f. As an example, suppose the 2D string is (x_1 x_2 $<$ x_3:x_4 $<$ x_5, x_2 x_3:x_4 $<$ x_1 x_5), where the notation x_3:x_4 indicates x_3 and x_4 are in the same set. We first construct the picture shown in Fig. 2.3, based upon 1D string u, by placing objects having the same spatial location (i.e., objects related by the ’=’ operator) in the same “slot”.

x1 x2	x3 x4	x5

FIGURE 2.3. Reconstruction based upon 1D string u.

Next, we utilize 1D string v to construct the final picture, as shown in Fig. 2.4.

x2	x3 x4	
x1		x5

FIGURE 2.4. Reconstruction based upon 1D string v.

If all the symbols in a 2D string are distinct, the reconstructed picture is unique. If, however, there are identical symbols in the 2D string, then in general there may be several different reconstructed pictures. For example, the 2D string ($a < a$,$a < a$) may represent a picture with a in both the upper-left and lower-right slots, or a picture with a in both the upper-right and lower-left slots. How to characterize such ambiguous pictures for different types of 2D strings is discussed in [2].

2.3 Picture matching

2D string representation provides a simple approach to perform subpicture matching on 2D strings. The *rank* of each symbol in a string u, which is defined to be one plus the number of '<' preceding this symbol in u, plays an important role in 2D string matching. We denote the rank of symbol b by $r(b)$. For example, symbols in the string "$ad < b < c$" have ranks 1, 1, 2, 3, respectively, and symbols in the string "$a < c$" have ranks 1, 2, respectively.

A substring where all symbols have the same rank is called a *local substring.*

A string α is *s-contained* in a string β, if α is a subsequence of a permutation string of β.

A string α is a *type-k 1D subsequence* of string β, if (a) α is s-contained in β, and (b) if $a_1 w_1 b_1$ is a substring of α, a_1 matches a_2 in v and b_1 matches b_2 in β, then

(type-0) $r(b_2)$-$r(a_2)$$\geq$$r(b_1)$-$r(a_1)$ or $r(b_1)$-$r(a_1)$=0;
(type-1) $r(b_2)$-$r(a_2)$$\geq$$r(b_1)$-$r(a_1)$>0 or $r(b_2)$-$r(a_2)$=$r(b_1)$-$r(a_1)$=0;
(type-2) $r(b_2)$-$r(a_2)$=$r(b_1)$-$r(a_1)$.

Now we can define the notion of type-k (i=0,1,2) 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-k 2D subsequence* of (u,v) if (a) u' is type-k 1D subsequence of u, and (b) v' is type-k 1D subsequence of v. If the above is true, we say f' is a *type-k sub-picture* of f.

In Fig. 2.5, $f1$, $f2$ and $f3$ are all type-0 $f1$ and $f2$ are type-1 The 2D string representations are:

f ($ad < b < c$, $a < bc < d$);
$f1$ ($a < b$, $a < b$);
$f2$ ($a < c$, $a < c$);
$f3$ ($ab < c$, $a < bc$).

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

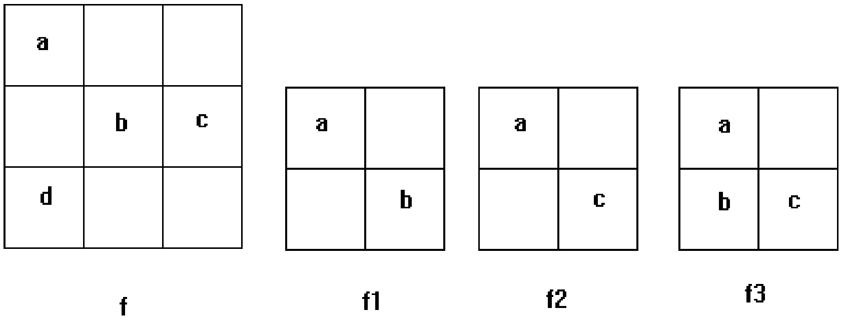


FIGURE 2.5. Picture matching examples, where $f1$ is a type-2 subpicture of f, $f2$ is a type-1 subpicture of f, and $f3$ is a type-0 subpicture of f.

In type-1 subsequence matching, each local substring in u should be matched against a local substring in v. For example, in Fig. 2.5 substring "a" in $a < c$ of $f2$ is a subsequence of "ad" in $ad < b < c$ of f, and substring "c" in $a < c$ is a subsequence of "c" in $ad < b < c$. Notice the skipping of a rank is allowed in type-1 subsequence matching. Therefore, the type-1 subsequence matching problem can be considered as a two-level subsequence matching problem, with level-1 subsequence matching for the local substrings, and level-2 subsequence matching for the "super-string" where each local substring is considered as a super-symbol, and super-symbol u_1 matches super-symbol v_1 if u_1 is a subsequence of v_1.

Type-2 subsequence matching is actually simpler, because the rank cannot be skipped. That is to say, if local substring u_1 of u matches local substring v_1 of v, then substring u_i of u must match substring v_i of v for any i greater than 1. In the example shown in Fig 2.5, $v =$ "$a < c$" of $f2$ is not a type-2 subsequence of "$ad < b < c$" of f.

2.4 Computer-aided design database

The 2D string representation is an efficient way to represent symbolic pictures, allowing an effective means for queries on image databases, spatial reasoning, visualization and browsing. At the same time, we note that the performance of the 2D string iconic indexing depends on the abstraction from segmented images to symbolic pictures [6]. Many researchers have been looking for good abstraction techniques from iconic images to symbolic representation [7, 8, 9, 10, 11]. The iconic indexing approach should be combined with pattern recognition so that iconic indices can be automatically created.

In the following we describe some typical applications of the Theory of

Symbolic Projection to image information retrieval. These are based upon papers by researchers from many different countries and are indicative of the diversity of potential applications.

A CAD database "2 cranes on the hull with a superstructure behind them and a mast, radar and funnel on the superstructure." To process such queries Hildebrandt and Tang applied the symbolic projection technique in 3D to symbolic voxel models [12].

CAD data is usually stored in one of two forms: the boundary representation or the constructive solid geometry form. In the boundary representation an object is segmented into non-overlapping faces. Each face is modelled by bounding edges and edges by end vertices. So the object is modelled by a tree of depth three. In constructive solid geometry there are primitives such as cylinders, boxes and cones combined and modified by operations such as union, intersection, difference, rotation and scale. The database catalog of the CAD database is assumed to contain a simple voxel model generated from the CAD data for queries.

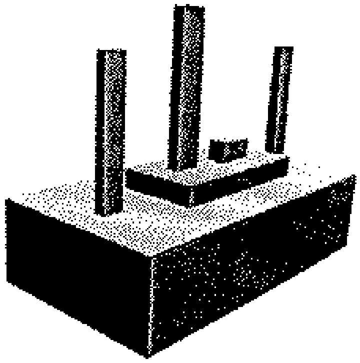
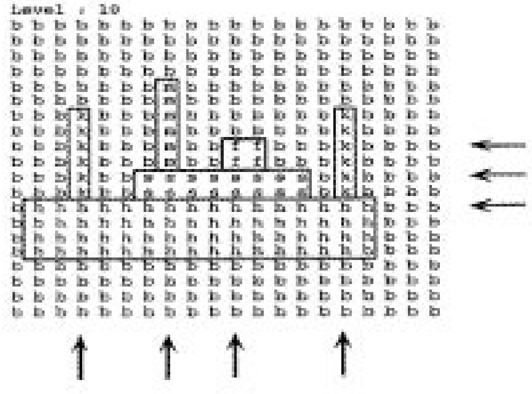


FIGURE 2.6. Voxel ship model.

GIS databases typically store a collection of co-registered two dimensional images of certain properties such as brightness, spot height and slope, together with vector data such as roads, river and contours. Layers of raster data can be interpreted directly as images and grouped together to give voxel data. Vector data would have to be first converted into low resolution raster data and then used as image or voxel data. The simplified 2D representation could then be used on each band in the GIS to index spatial information, and it may be possible to use the three dimensional form to index spatial relations between all bands.

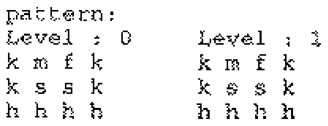
The simple voxel model encodes a three-dimensional object into slices. An example is illustrated in Fig. 2.6, which shows a 20 by 20 by 20 voxel model. Such models could be used to index a collection of CAD models and would allow 3D spatial queries. A slice through this model is shown in Fig. 2.7,



```
Level : 10
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b m b b b b b b b b b b b b
b b b b b b b m b b b b b b b b b b b b
b b b k b b b m b b b b b b b k b b b b
b b b k b b b m b b b b b b b k b b b b
b b b k b b b m b b f f b b b k b b b b
b b b k b b b m b b f f b b b k b b b b
b b b k b b s s s s s s s s b k b b b b
b b b k b b s s s s s s s s b k b b b b
b h h h h h h h h h h h h h h h h b b b
b h h h h h h h h h h h h h h h h b b b
b h h h h h h h h h h h h h h h h b b b
b h h h h h h h h h h h h h h h h b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
b b b b b b b b b b b b b b b b b b b b
```

FIGURE 2.7. Slice of voxel model: h=hull, s=superstructure, k=kingpost, m=mast and f=funnel. The arrows indicate a single type-1 match returned for the query pattern of Fig. 2.8.

where the letters 'h', 's', 'k', 'm', and 'f' denote "hull", "superstructure", "kingpost", "mast" and "funnel", respectively.



```
pattern:
Level : 0      Level : 1
k m f k        k m f k
k s s k        k s s k
h h h h        h h h h
```

FIGURE 2.8. A query pattern.

In Fig. 2.8, a 3 by 4 by 2 pattern for a type-1 query is shown. A single type-1 match returned is indicated by the arrows in Fig. 2.7. This search pattern would be used with query "Find ship with 2 kingposts above hull with superstructure in between them and mast followed by funnel above superstructure".

A graphical user interface was constructed for a prototype ship database application (Fig. 2.9). For textual information associated with a ship, typical database forms were employed. To enter and display the spatial relations in the database, a graphical interface was employed where icons representing objects could be placed on the grided outline of a hull, viewed from above (top of Fig. 2.9). From this input, the required spatial relations could be determined and placed in a relational table to perform the database query. In addition to type-k queries, the system also supports pairwise relations matching so that similar patterns can be found more efficiently.

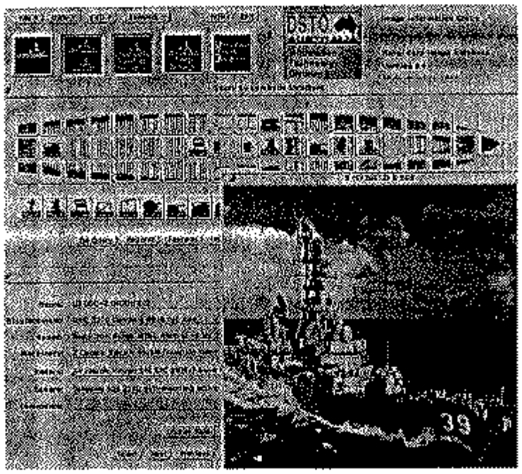


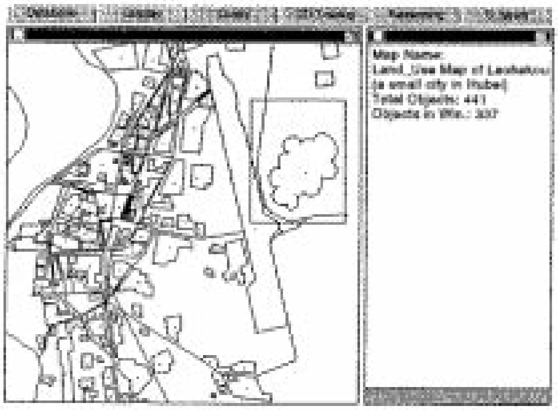
FIGURE 2.9. Graphical user interface combining textual and symbolic query with image retrieval.

2.5 Geographical information systems

The application of Theory of Symbolic Projection to Geographical Information Systems was studied by Yuguo Sun [13].

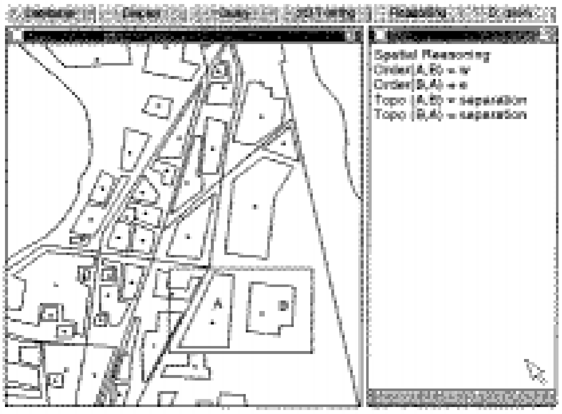
Sun generalized the 2D G-string [5] to the 2D T-string. The 2D T-string is able to represent three different types of qualitative spatial relations, i.e. topological relations, ordering relations and auxiliary relations. The topological relations describe local spatial relations, such as equal, disjoint, meet, edge, contain, and partial overlap. The ordering relations are the two basic global spatial relations $<$ and $|$ (edge-to-edge concatenation). The auxiliary relations include surround, partially surround, and quasi-partially surround. The cutting mechanism basically follows the cutting mechanism for the G-string, but refined by additional rules so that only the necessary cutting lines important to one of the above types of relations will be drawn. Techniques for constructing the 2D T-string from the symbolic picture, and for spatial reasonings, have been developed. Since they are similar to the techniques described in [14], the details will not be presented here.

An experimental Spatial Relations Retrieval System based upon the 2D T-String was implemented for geographical information retrieval. The system supports spatial reasoning, basic spatial relation query, complex spatial relation query and similarity query. The geographical data is the land use map of the Laohekou City in Hubei Province of China. The land use map is shown in Fig. 2.10. The map contains 441 objects, and 337 of them are displayed in the window area. Fig. 2.11 is an example of spatial reasoning. The user selects a rectangular area of interest. The ordering relations ("A is to the west of B", and "B is to the east of A") and topological relations



Map Name:
Land_Use Map of Laohekou
(a small city in Hubei)
Total Objects: 441
Objects in Win.: 337

FIGURE 2.10. Land use map.



Spatial Reasoning
Order(A,B) = w
Order(B,A) = e
Topo (A,B) = separation
Topo (B,A) = separation

FIGURE 2.11. Example of spatial reasoning.

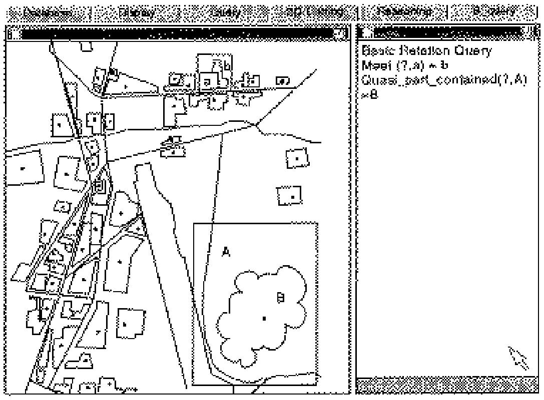
("A and B are separated spatially") can be derived from the 2D T-string and displayed in the window area on the right.

Fig. 2.12 illustrates basic spatial relation query. The system can find out object B meets object A, and object B is quasi-part-contained in A.

Fig. 2.13 illustrated complex spatial relation query. The system can find out object D is contained in A and to the west of E, and object D is contained in A and to the northwest of C.

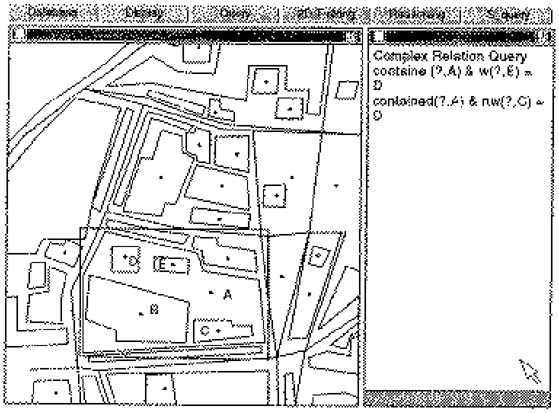
Fig. 2.14 illustrates similarity query. The query is shown in the upper right window, where A and B are resident land, and C is the railway, and their approximate spatial relations are as shown. The result is displayed in the lower right window.

For similarity retrieval, time-consuming graph matching is required. How-



Basic Relation Query
Meet (?,a) = b
Quasi_part_contained(?,A) =B

FIGURE 2.12. Basic spatial relation query.



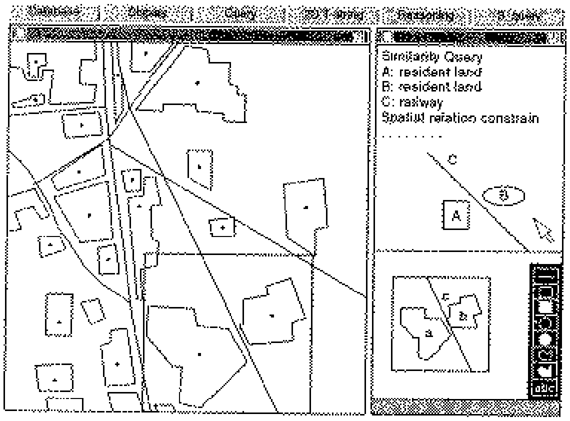
Complex Relation Query
contains (?,A) & w(?,E) = D
contained(?,A) & nw(?,C) = D

FIGURE 2.13. Complex spatial relation query.

ever, in practical applications, the targets are restricted to a prespecified small window area, and the constraints include not only spatial constraints, but also constraints on the objects' attribute values such as shape, color, etc. Therefore, similarity retrieval can be computed in a reasonable time.

2.6 Retrieval of similar Chinese characters

Although many methods have been proposed to solve the problem of Chinese character retrieval, the problem of retrieval spatially similar Chinese characters still remains. There are several motivations to consider the retrieval of similar Chinese characters. First, it can be useful in learning Chinese characters. The structurally similar Chinese characters can be re-

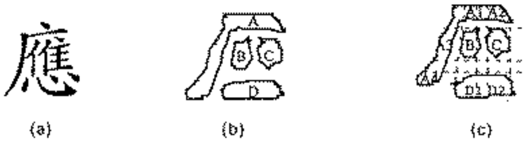


Similarity Query
A: resident land
B: resident land
C: railway
Spatial relation constrain
....

FIGURE 2.14. Similarity query.

trieved and presented to the student, so that the student can remember the components of the characters and their meanings. Second, similarity retrieval is also useful for Chinese character recognition, because it is capable of clustering similar characters.

Chang and Lin applied the Symbolic Projection Theory to Chinese character retrieval [15], by regarding the Chinese character as a symbolic picture. As illustrated in Fig. 2.15(a), the original image corresponds to a Chinese character. Pattern recognition algorithm can be applied to segment the image into four major components A, B, C and D, as illustrated in Fig. 2.15(b).



(a) (b) (c)

FIGURE 2.15. The original Chinese character (a), the symbolic picture (b) and the segmented symbolic picture (c).

The technique of orthogonal relations [16] can then be applied to discover the important orthogonal relations and convert the symbolic picture into the 2D string. As illustrated in Fig. 2.15(c), the following orthogonal relations are discovered:

$Ortho-relation$(B,A) = {A1,A3 }
$Ortho-relation$(C,A) = {A2,A3 }
$Ortho-relation$(B,D) = {D1 }
$Ortho-relation$(C,D) = {D2 }

Therefore, A is segmented into four pieces, and D is segmented into two pieces. The 2D string is (A4 < A3 < A1 B D1 < A2 C D2, A4 D1 D2 < A3 B C < A1 A2). Given a Chinese character, it can be transformed into the 2D string and then matched against the 2D strings of other Chinese characters. By using type-0, type-1 or type-2 matching and a cost algorithm, Chang and Lin can find weakly similar, partially similar or strongly similar Chinese characters.

In a related application, 2D strings have been applied to the retrieval and recognition of handwritten signatures [17].

2.7 Three-dimensional image database querying

An extension of 2D strings to deal with three dimensional imaged scenes was proposed in [18]. The approach relies on the consideration that two-dimensional iconic queries and 2D string-based representations are effective for the retrieval of images representing 2D objects or very thin 3D objects, but they might not allow an exact definition of spatial relationships for images representing scenes with 3D objects. In fact in this case, an incorrect representation of the spatial relationships between objects may result due to two distinct causes. First, 2D icons cannot reproduce scene depth. 2D icon overlapping can be used only to a limited extent since it impacts on the understandability of the query. Second, as demonstrated by research in experimental and cognitive psychology, the mental processes of human beings simulate the physical world processes. Computer generated line drawings representing 3D objects are regarded by human beings as 3D structures and not as image features, and they imagine spatial transformations directly in 3D space.

Therefore, an unambiguous correspondence is established between the iconic query and image contents, if the spatial relationships referred to are those between the objects in the scene represented in the image, rather than those between the objects in the image. The dimensionality of data structures associated with icons must follow the dimensionality of the objects in the scene represented in the image. A 3D structure should be employed for each icon to describe a 3D scene. An example is illustrated in Fig. 2.16.

Representations of images are derived considering 3D symbolic projections of objects in the 3D imaged scene. Thirteen distinct operators, corresponding to the interval logic operators, distinguish all the possible relationships between the intervals corresponding to the object projects on each axis.

Retrieval systems employing this ternary representation of symbolic projections have been expounded in [18] and [19]. In these approaches, the user reproduces a three-dimensional scene by placing 3D icons in a virtual

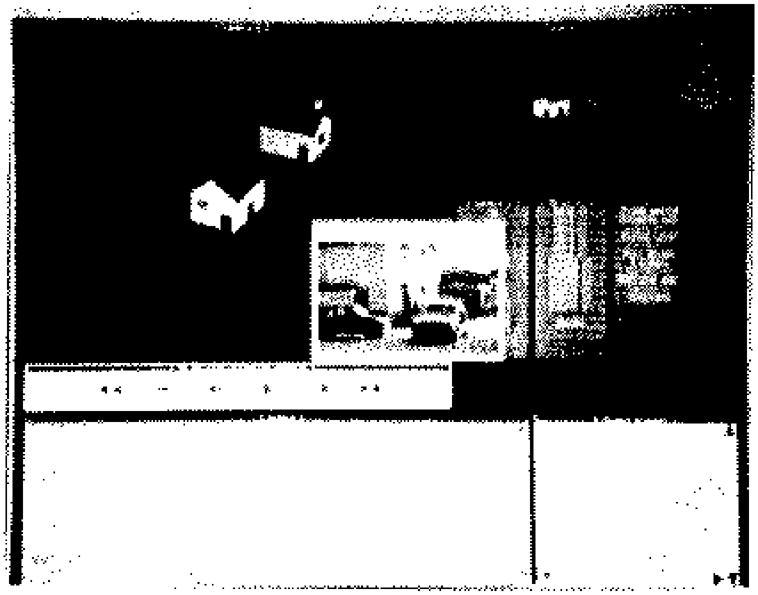


FIGURE 2.16. Querying a three-dimensional scene using pairwise 3D relations.

space and sets the position of the camera in order to reproduce both the scene and the vantage point from which the camera was taken. A spatial parser translates the visual specification into the representation language and retrieval again is reduced to a matching between symbolic strings.

2.8 Medical image database system

2D string has been used in recognizing fungi in medical research [20]. This section describes the incorporation of 2D strings in a medical image database system.

Radiological examinations are extremely important in health care. X-ray film is the medium conventionally used for medical image archival purposes. A PACS (Picture Archiving and Communication System) computer system that supports digital image handling in a hospital environment. Facilities typically provided by a PACS include image entry, archiving, communication, presentation, etc. A PACS is connected by high-speed network with the HIS (Hospital Information System), in order to handle textual patient data together with images.

Such new environment opens a whole new world of possibilities for the utilization of medical images in the clinical environment, including computer assisted diagnosis, rediotherapy planning, surgery planning, medical training, etc. Medical image indexing and retrieval by content, in particular, play a special role in this net setting.

I^2C is an image database system which has been developed as a platform for the design, implementation and evaluation of medical image indexing and retrieval by content schemes [21]. This system allows the user to define regions of interest (ROI) on the query image, and adjust the relative importance of different regions as well as their characteristics. The user can draw a sketch and adjust the search parameters, to direct the image retrieval process. The main concept in the design of I^2C are image classes and image description types. An image class encapsulates algorithms for the organization, processing and indexing of the images in it. When a request for retrieval is placed with I^2C, it is directed to the appropriate class. The concept of the image description type encapsulates all the details of an indexing and retrieval by content scheme, including the use of 2D strings.

References

[1] S. K. Chang, "The Reconstruction of Binary Patterns from Their Projections," *Communications of the ACM* **14**, 21-25 (1971).

[2] S. K. Chang, Q.Y. Shi and C.W. Yan, "Iconic Indexing by 2D Strings," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-9**, 413-428 (1987).

[3] S. K. Chang, E. Jungert and Y. Li, "Representation and Retrieval of Symbolic Pictures Using Generalized 2D Strings," in *Proc. of SPIE Visual Communications and Image Processing Conference*, pp. 1360-1372, (1989).

[4] E. Jungert, "Extended Symbolic Projections as a Knowledge Structure for Spatial Reasoning and Planning," in *Pattern Recognition*, (J. Kittler, Ed.), (Springer Verlag, Berlin and Heidelberg), pp. 343-351, (1988).

[5] S. K. Chang and E. Jungert, *Symbolic Projection for Image Information Retrieval and Spatial Reasoning*, (Academic Press, London), 1996.

[6] S. K. Chang, C. W. Yan, T. Arndt and D. Dimitroff, "An Intelligent Image Database System," *IEEE Trans. on Software Engineering, Special Issue on Image Database*, 681-688 (1988).

[7] S. K. Chang and S.H. Liu, "Indexing and Abstraction Techniques for Pictorial Databases," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**, pp. 475-484 (1984).

[8] S. L. Tanimoto, "An Iconic/Symbolic Data Structuring Scheme," *Pattern Recognition and Artificial Intelligence*, (Academic Press, New York), pp. 452-471, (1976).

[9] L.G. Shapiro and R.M. Haralick, "A Spatial Data Structure", *Technical Report CS 79005-R*, Dept. Comput. Sci., Virginia Polytech. Inst. and State University, (1979).

[10] M. Sties, B. Sanyal and K. Leist, "Organization of Object Data for an Image Information System," in *Proceedings 3rd Int. Joint Conference on Pattern Recognition*, pp. 863-869, (1976).

[11] A. Klinger, M.L. Rhode and V.T. To, "Accessing Image Data," *Int. J. Policy Analysis Information Syst.* **1**, pp. 171-189 (1978).

[12] J.W. Hildebrandt and K. Tang, "Symbolic two and three dimensional picture retrieval," in *Workshop on Two and Three Dimensional Spatial Data: Representation and Standards*, (Australian Pattern Recognition Society, Perth, WA.), (1992).

[13] Yu Guo Sun, *Description of Topological Spatial Relations and Representation of Spatial Relations using 2D T-String*, Ph.D. Thesis, Wuhan Surveying Technology University, Wuhan, China, 1993.

[14] S. Y. Lee and F. J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images Using 2D C-string Knowledge Representation," *Pattern Recognition* **25**, 305-318, (1992).

[15] C. C. Chang and D. C. Lin, "Utilizing the Concept of Longest Common Subsequence to Retrieve Similar Chinese Characters," *Computer Processing of Chinese and Oriental Languages* **8**, pp. 177-191, (1994).

[16] Chang, S. K. and E. Jungert, "A Spatial Knowledge Structure for Visual Information Systems," in *Visual Languages and Applications*, (T. Ichikawa, R. Korfhage and E. Jungert, Eds.), pp. 277-304, (Plenum Publishing Co, New York), 1990.

[17] Sethi, I. K. and K. Han, "Use of Local Structural Association for Retrieval and Recognition of Signature Images," in *SPIE Proceedings of Storage and Retrieval for Image and Video Databases III*, (W. Niblack and R. C. Jain, Eds.), **2420**, pp. 125-134, San Jose, February 9-10, (1995).

[18] A. Del Bimbo, M. Campanai and P. Nesi, "A Three-Dimensional Iconic Environment for Image Database Querying," *IEEE Transactions on Software Engineering* **19**, pp. 997-1011 (1993).

[19] Bimbo, A. Del, E. Vicario, and D. Zingoni, "A Spatial Logic for Symbolic Description of Image Contents," Journal of Visual Languages and Computing, **3**, pp. 267-286, (1994).

[20] M. L. Dorf, A. F. Mahler and P. F. Lehmann, "Incorporating Semantics into 2D Strings," in *Proceedings of 22nd Annual ACM Computer Science Conference*, (D. Cizmar, Ed.), Phoenix, AZ, pp. 110-115, (1994).

[21] S. Kostomanolakis, M. Lourakis, C. Chronaki, Y. Kavaklis and S. C. Orphanoudakis, "Indexing and Retrieval by Pictorial Content in the i^2C Image Database System," *Technical report, Institute of Computer Science, Foundation for Research and Technology-HELLAS*, Heraklion, Crete, Greece, (1993).