To appear in the *Journal of Visual Languages and Computing*, Special Issue on Visual Query Systems, March 1995.

A Pictorial Query-By-Example Language

Dimitris Papadias* and Timos Sellis

Computer Science Division, Department of Electrical and Computer Engineering,
National Technical University of Athens, Zographou Campus, Greece 15773.

A *symbolic image* is an array representing a set of objects and a set of spatial relations among them. Symbolic images and related structures have been used in a number of applications including Image Databases, Spatial Reasoning, Path Planning and Spatial Pattern Matching. In this paper we describe a *Pictorial Query-By-Example* (PQBE) language aimed at the retrieval of *direction relations* from symbolic images. As in the case of verbal Query-By-Example, PQBE generalises from the example given by the user, but instead of having queries in the form of skeleton tables showing the relation scheme, we have skeleton images which are themselves symbolic images. PQBE provides an intuitive interface for use in geographic applications because of its close correspondence with the structure of space.

1. Introduction

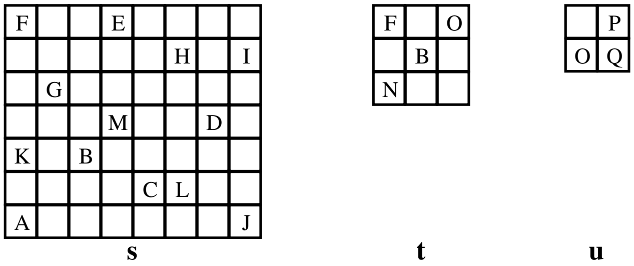
The topic of the paper is the retrieval of *direction relations* stored in symbolic images using a visual language. Direction relations deal with order in space (e.g., *North, East, NorthEast, SameLevel*) and their representation and processing is important for Geographic Applications where the formalization of spatial relations is crucial for user interfaces and query optimisation strategies. A survey and an experimental study regarding the use of direction relations at geographic scales can be found in [1]. Furthermore, the importance of direction relations has been pointed out by several researchers in areas including Geographic and Multimedia Databases [2,3], Spatial Data Structures [4], Cognitive Science [5] and Linguistics [6].

A *symbolic image* is an array representing a set of objects and a set of direction relations among them. Symbolic images and related structures such as the *2D strings* [7] (one dimensional encodings of symbolic images), *symbolic arrays* [8] (hierarchical symbolic images) and *symbolic spatial indexes* [9] (symbolic images where the symbols correspond to direction and topological representative points) have been used in applications including Context-Based Retrieval in Image Databases [10], Spatial Reasoning [11], Path Planning [12] and Image

---

[*] Correspondence should be addressed to Dimitris Papadias, Dept. of Computer Science and Engineering, University of California at San Diego, La Jolla, California, USA 92093-0114, dimitris@cs.ucsd.edu.

Similarity Retrieval [13]. For the following examples we will consider that our database consists of the symbolic images of Figure 1; these images could correspond to visual scenes, geographic maps or other forms of spatial data.



F			E				
					H		I
	G						
			M			D	
K		B					
				C	L		
A							J

s

F		O
	B	
N		

t

	P
O	Q

u

Fig. 1 Symbolic images

Using symbolic images, such as the previous ones, we can represent nine binary, pairwise disjoint, direction relations: {NorthWest, RestrictedNorth, NorthEast, RestrictedWest, SamePosition, RestrictedEast, SouthWest, RestrictedSouth, SouthEast}. These relations have been called *primitive direction relations* [14] and correspond to the highest resolution that we can achieve using one symbol per object in symbolic images. The primitive relations are transitive and *SamePosition* is also symmetric. The rest form four pairs of converse relations (e.g., NorthEast(X,Y) ⇔ SouthWest(Y,X)). Exactly one of the previous relations holds true between any pair of objects (SamePosition is true for the pair X, X). Figure 2 illustrates the primitive relations among object symbols in a symbolic image. The symbol Y denotes the *reference* object and the other symbols refer to the direction relation depending on the position of the *primary* object in the symbolic image (the term *primary object* denotes the object to be located and the term *reference object* denotes the object in relation to which the primary object is located).

NorthWest	RestrictedNorth	NorthEast
RestrictedWest	**Y**	RestrictedEast
SouthWest	RestrictedSouth	SouthEast

Fig. 2 Direction relations encoded in symbolic images

According to our interpretation, symbolic images are just a way of representing direction constraints and *they do not preserve absolute metric information but only order in space*. As a consequence the information preserved in an image does not change if we add or remove empty rows and columns. Notice that the same pair of objects may exist in more than one images (e.g., objects F and B); in such cases we assume that the images are consistent, i.e., they both preserve the same relationship between the objects. A discussion about the construction of symbolic images from input image representations and consistency issues can be found in [14].

In this paper we describe a Pictorial Query-By-Example (PQBE) language aimed at the retrieval of the previous direction relations from symbolic images. As in the case of verbal Query-By-Example [15], PQBE generalises from the example given by the user, but instead of having queries in the form of skeleton tables showing the relation scheme, we have *skeleton images* (otherwise called *skeleton arrays*) which are themselves symbolic images. It has been argued in [15] that QBE is easier to learn by novice users. Similarly we believe that PQBE is easier to learn and use than other query languages that deal with space, such as extensions of SQL [16,17], as well as verbal QBE, because of its close correspondence with the structure of the 2D space.

Chang et al. [7,10] developed a similar pictorial language for retrieving *object configurations* stored in 2D strings. In addition to the retrieval of object configurations, PQBE has the ability to express *negation*, *independent sub-conditions* in the form of multiple skeleton images, *union*, *intersection* and *join* operations, *image and relation retrieval,* and to perform inference. Furthermore, with the incorporation of some additional features, PQBE can be used as an easy to use GIS (Geographic Information System) interface, independently of the underlying representation. For instance, instead of relations between objects in symbolic images, the queries described in the following sections could refer to direction relations between cities in digital maps or similar application domains.

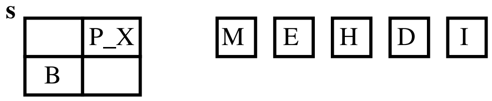
A short version of the paper appears in [18]. Although the papers contain similar examples, this paper extends the previous one by formally describing the results of queries and by including two additional sections about inference and update operations. In the rest of the paper we present the capabilities of PQBE by providing several example queries. Section 2 describes queries that involve primary object retrieval using one skeleton image. Section 3 discusses queries that use multiple skeleton images; such queries are useful in expressing independent sub-conditions, relation retrieval, join, union and intersection. Section 4 is concerned with the retrieval of entire images that satisfy certain spatial conditions. Section 5 describes how inference is achieved in PQBE using composition of spatial relations. Section 6 describes update operations and Section 7 concludes with a discussion about further extensions.

2. Queries Involving One Skeleton Image

This class of queries involves the retrieval of sub-images that contain primary objects which satisfy some spatial constraints with respect to reference object constants or variables. Variables denoting objects and images are preceded by the character "_", while constants appear without qualification. In our notation, primary objects are preceded with the character *P* (printing character), i.e., the character P before an object variable or constant causes its value to be retrieved and displayed.

Each symbolic and skeleton image I can be described by two sets: the set O(I) that contains the objects of I and the set C(I) that contains the primitive direction relations (constraints) between all pairs of objects in I. For image u, for instance, O(u)={O,P,Q} and C(u) = {RestrictedEast(Q,O), SouthWest(O,P), RestrictedNorth(P,Q)} (for simplicity we do not include *SamePosition* and the converse relations in the set C). The result of a query is the set of all symbolic (sub)images[1] that satisfy the spatial conditions imposed by the sets O and C representing skeleton images. For each of the following example queries we will include the skeleton image, the imposed spatial constraints and the resulting (sub)images with respect to the database of Figure 1. It is important to emphasise that all binary constraints among the domain variables and the constants that appear in a skeleton image must be taken into account when processing the query. Furthermore, for the following queries we presume the *closed world*, the *domain closure* and the *unique name* assumptions (although these assumptions are not always realistic for geographical applications).

The query in the skeleton image of Figure 3 will retrieve the sub-images of s (image constant) that contain an object X (primary object variable) where X is *NorthEast* of B (reference object constant) in s. The result is the set of all symbolic (sub)images I that are described by the following sets: $O(I)=\{X \mid NorthEast(X,B)\in C(s)\}$[2] and $C(I)=\{\}$. The sub-images of s that satisfy the previous sets are also illustrated in Figure 3 in a *south to north* and *west to east* order.



Skeleton Image Results

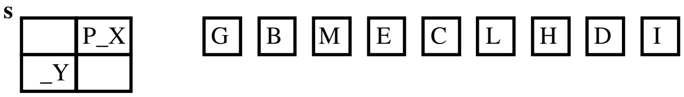
Fig. 3 Query involving one primary object and one reference object constant

Similarly, the query in the skeleton image of Figure 4 will retrieve the sub-images I of s that contain an object X where X is *NorthEast* of some object Y (reference object variable) in s, i.e., $O(I)=\{X \mid \exists Y(NorthEast(X,Y)\in C(s)\}$ and $C(I)=\{\}$. The result of this query is a superset of the previous result, because the sub-images containing objects *NorthEast* of B are also included. Notice that for both of the above queries, as for all the queries that result in sub-images containing a single object, the set C(I) is empty, since there are no binary constraints.

---

[1] A symbolic image I is a sub-image of J iff: $O(I)\subseteq O(J)$ and $C(I)\subseteq C(J)$.

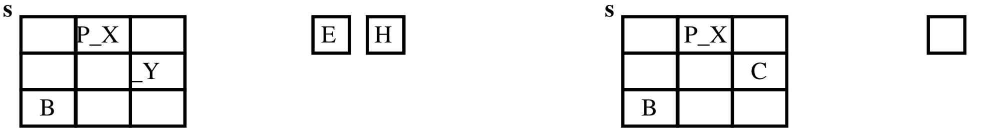
[2] The complete description for the set O(I) is $O(I)=\{X \mid X\in O(s) \wedge NorthEast(X,B)\in C(s)\}$. For brevity, we omit $X\in O(s)$ for the following descriptions because it is redundant; if $NorthEast(X,B)\in C(s)$, then it must be true that $X\in O(s)$.



Skeleton Image Results

Fig. 4 Query involving one primary object and one reference object variable

We can also have queries with more than one primary, or reference, objects. The query in the skeleton image of Figure 5a, for instance, will retrieve all sub-images of s containing an object X that is *NorthEast* of B and *NorthWest* of some object Y, where Y is *NorthEast* of B in s. The result is the set of all symbolic (sub)images I that are described by the sets: $O(I)=\{X \mid NorthEast(X,B)\in C(s) \wedge \exists Y(NorthWest(X,Y)\in C(s) \wedge NorthEast(Y,B)\in C(s))\}$ and $C(I)=\{\}$. Notice that the result is a subset of the objects that are retrieved through the query of Figure 3, because the primary objects in this case must satisfy more specialised spatial conditions. The query in the skeleton image of Figure 5b will retrieve all sub-images of s containing an object X that is *NorthEast* of B and *NorthWest* of C, where C is *NorthEast* of B in s. The query is similar to the previous one, but variable Y is instantiated to object C, i.e., $O(I)=\{X \mid NorthEast(X,B)\in C(s) \wedge NorthWest(X,C)\in C(s) \wedge NorthEast(C,B)\in C(s)\}$ and $C(I)=\{\}$. The result is empty since C is not *NorthEast* of B in s.



Skeleton Image Results Skeleton Image Results

a **b**

Fig. 5 Queries involving two reference objects

PQBE also allows "negative" reference objects. For example, the query in the skeleton image of Figure 6 will retrieve all sub-images of s that contain an object X which is *NorthEast* of B and there does not exist an object Y such that X is *NorthWest* of Y and Y is *NorthEast* of B in s. From the set of sub-images containing objects *NorthEast* of B (Figure 3), this query will retrieve only the subset of sub-images which are not retrieved through the query of Figure 5a, i.e., $O(I)=\{X \mid NorthEast(X,B)\in C(s) \wedge \sim\exists Y(NorthWest(X,Y)\in C(s) \wedge NorthEast(Y,B)\in C(s))\}$ and $C(I)=\{\}$.

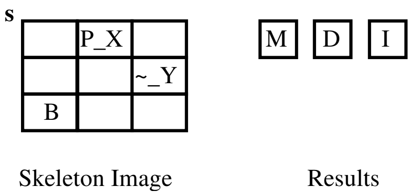


Fig. 6 Query involving a negative reference object

Unlike the previous queries which refer to image constants, the queries of Figure 7 refer to image variables. The query in the skeleton image of Figure 7a, for instance, will retrieve the sub-images that contain an object X which is *NorthEast* of B in some image J (image variable): $O(I)=\{X \mid \exists J\ (NorthEast(X,B) \in C(J))\}$ and $C(I)=\{\}$. The result is a superset of the sub-images retrieved by the query of Figure 3, because it contains the objects *NorthEast* of B in image s, as well as, the objects *NorthEast* of B in other images (e.g., object O). On the other hand the query of Figure 7b will retrieve the complement of the previous query, i.e., all sub-images I that do not contain an object which is *NorthEast* of B in some image J: $O(I)=\{X \mid \sim\exists J(NorthEast(X,B) \in C(J)\}$ and $C(I)=\{\}$.

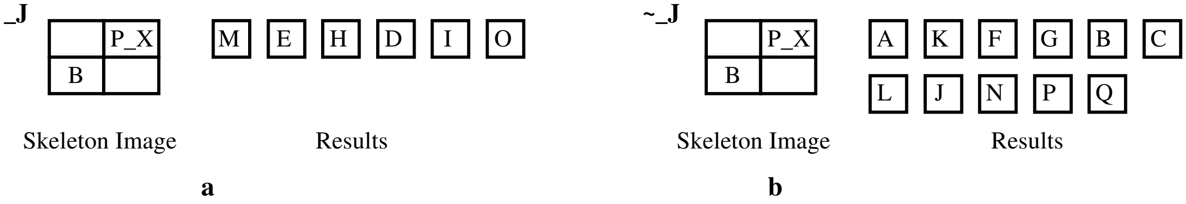


Fig. 7 Queries involving image variables

We can also have as results of queries, sub-images that do not contain single objects, but configurations of two or more objects. The query in the skeleton image of Figure 8, for instance, will retrieve all sub-images I containing objects X, Y and W and Z that satisfy the following constraint: $O(I)=\{X,Y,W,Z \mid \exists J\ (SouthWest(X,W) \in C(J) \wedge RestrictedWest(X,Z) \in C(J) \wedge NorthWest(X,Y) \in C(J) \wedge NorthWest(W,Z) \in C(J) \wedge RestrictedNorth(W,Y) \in C(J) \wedge NorthEast(Z,Y) \in C(J))\}$ and $C(I)=\{SouthWest(X,W), RestrictedWest(X,Z), NorthWest(X,Y), NorthWest(W,Z), RestrictedNorth(W,Y), NorthEast(Z,Y)\}$.

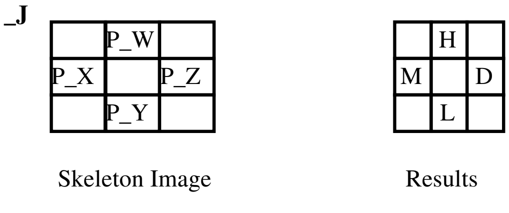


Fig. 8 Query involving retrieval of object configurations

PQBE is a flexible way to express queries involving direction relations and especially queries that involve complex spatial conditions. Since all binary constraints among the domain variables and constants that appear in a skeleton image must be taken into account when processing the query, the total number of binary direction constraints that we should express in a verbal query language (such as SQL or QBE) is $(n-1)+(n-2)+\ldots+1=n(n-1)/2$, where n is the total number of objects (we omit SamePosition and the converse relations). For the query of Figure 8, for instance, we have four objects and therefore we need six lines of QBE. In addition to being more economical, PQBE is more intuitive because of its close correspondence with the structure of space.

3. Queries Involving Multiple Skeleton Images

This section describes queries that involve more than one skeleton images. Such queries are useful in the retrieval of direction relations between object constants or variables. For instance, the result of the query of Figure 9a is a sub-image I of s describing the relation between A and F: $O(I)=\{A,F\}$ and $C(I)=\{R(A,F) \mid R(A,F) \in C(s)\}$. Similarly, the second query will generate the sub-image of s describing the relation among A, F and B, i.e., $O(I)=\{A,F,B\}$ and $C(I)=\{R_1(A,F), R_2(F,B), R_3(B,A) \mid R_1(A,F)\in C(s) \wedge R_2(F,B)\in C(s) \wedge R_3(B,A)\in C(s)\}$. Notice that the order of the skeleton images (in case that we have more than one) does not affect the result.

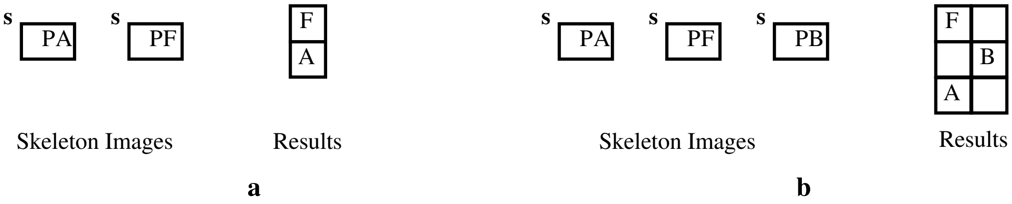


Fig. 9 Queries involving retrieval of relations between object constants

Queries with multiple skeleton images may also involve object variables. For instance, the query of Figure 10a will retrieve all the sub-images of u describing binary constraints between object Q and the other objects of u (we omit SamePosition(Q,Q)), i.e., $O(I)=\{Q, X \mid X\in O(u)\}$ and $C(I)=\{R(Q,X) \mid X\in O(I) \wedge R(Q,X)\in C(u)\}$. The query in Figure 10b will generate all the possible sub-images containing objects A and X, where X is *SouthEast* of A in image s. The sub-images that belong to the result are described by the following sets: $O(I)=\{A,X| \text{SouthEast}(X,M)\in C(s)\}$ and $C(I)=\{R(A,X)| X\in O(I) \wedge R(A,X)\in C(s)\}$.

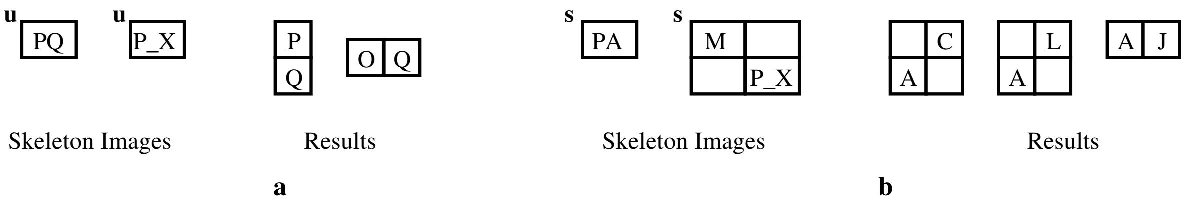
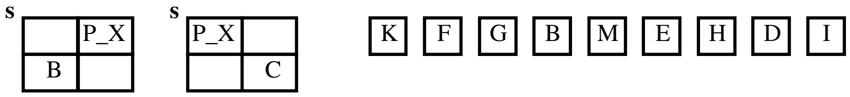


Fig. 10 Queries involving relation retrieval between object constants and variables

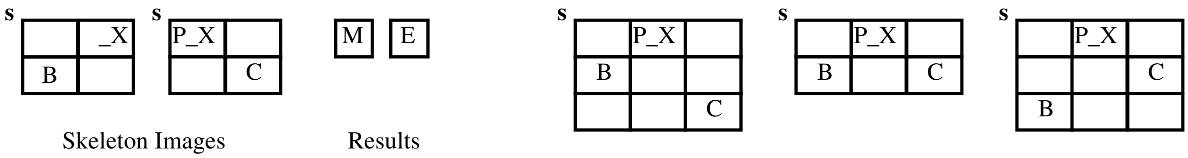
Using multiple skeleton images we can also express *union*, *intersection* and *join*. The query in Figure 11, for instance, will retrieve all sub-images I of s containing objects that are *NorthEast* of B, or *NorthWest* of C, i.e., the query expresses union: O(I)=$\{X|$ NorthEast(X,B)$\in$C(s)$\}$ $\cup$ $\{X|$ NorthWest(X,C)$\in$C(s)$\}$ and C(I)=$\{\}$. Unlike QBE, in PQBE all the variables to be printed should have the same name in order to express union, because otherwise the result would consist of a set of sub-images containing two different objects one of which is *NorthEast* of B and the other *NorthWest* of C in image s.



Skeleton Images Results

Fig. 11 Query involving union

The query in the skeleton image of Figure 12a will retrieve the sub-images of s containing objects that are *NorthEast* of B and *NorthWest* of C in image s, i.e., the result is the intersection of sub-images that satisfy these two conditions: O(I)=$\{X|$ NorthEast(X,B)$\in$C(s)$\}$ $\cap$ $\{X|$ NorthWest(X,C)$\in$C(s)$\}$ and C(I)=$\{\}$. Notice that in order to express this query using one skeleton image, we need to know the relation between B and C; otherwise the result will be empty as in Figure 5b. The formation for intersection in Figure 12a may be confusing because of its similarity with union since the queries of Figures 11 and 12a differ only in one printing character (the same problem exists in QBE). Figure 12b illustrates an alternative way to express the same query using union[3]. The idea is that we admit that the exact relation between B and C is unknown and that either B is *NorthWest* of C, or B is *RestrictedWest* of C or B is *SouthWest* of C. Out of the three skeleton images only one will produce the result (in particular the first one because B is *NorthWest* of C in s) while the rest do not have any contribution.



Skeleton Images Results

a

b

Fig. 12 Query involving intersection

Join is expressed by a chain of common objects connecting two or more skeleton images. The query in the skeleton image of Figure 13, for instance, will retrieve the sub-images of s containing an object X that is *NorthWest* of an object Y in image s, where Y is *NorthEast* of B in some image J: $O(I)=\{X \mid \exists Y \exists J\ (NorthWest(X,Y) \in C(s) \wedge NorthEast(Y,B) \in C(J)\}$ and $C(I)=\{\}$. This query corresponds to the join operation, where Y is the common object connecting the images. In particular, the result consists of all sub-images containing an object X which, in image s, is *NorthWest* of some of the objects retrieved through the query of Figure 7a (i.e., M, E, D, H, I, O).

---

[3] This way of expressing intersection was pointed out by Max Egenhofer and it is based on the concept of composition (see Section 5). That is, if B is *SouthWest* of X, and X is *NorthWest* of C, then the possible relations between B and C are *NorthWest*, *RestrictedWest,* or, *SouthWest.*

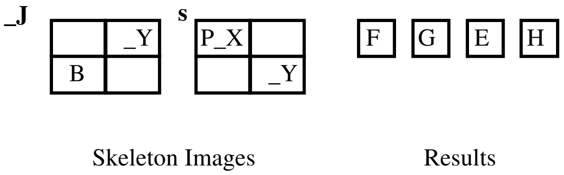


Fig. 13 Query involving join

Although most of the previous examples refer to single symbolic images, we can extend our queries for multiple images in a straightforward manner. The query in the skeleton images of Figure 14a, for instance, will generate all sub-images containing an object X that is *NorthWest* of B in image s or *NorthWest* of B in image t: $O(I)=\{X \mid NorthWest(X,B) \in C(s)\} \cup \{X \mid NorthWest(X,B) \in C(t)\}$ and $C(I)=\{\}$. On the other hand the query in the skeleton images of Figure 14b, will retrieve all sub-images containing an object X that is *NorthWest* of B in both images s and t: $O(I)=\{X \mid NorthWest(X,B) \in C(s)\} \cap \{X \mid NorthWest(X,B) \in C(t)\}$ and $C(I)=\{\}$.

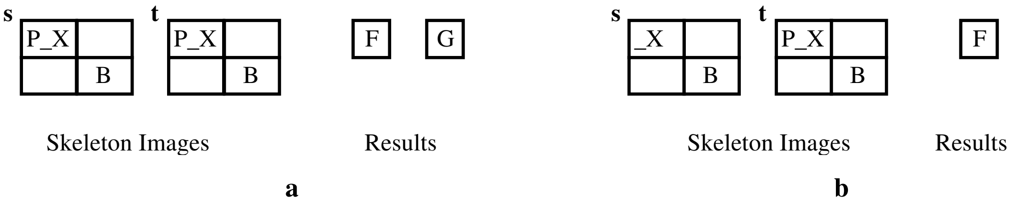
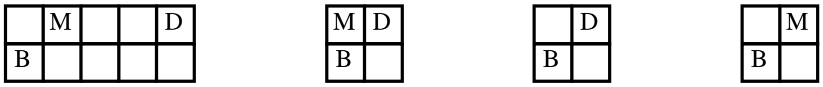


Fig. 14 Queries involving two images

Query-by-Pictorial Example (QPE) [19], is a relational query language for manipulating spatial and conventional relations in image databases. QPE allows the expression of spatial queries by pointing on the display screen using special input devices. Despite the name similarity the concept of QPE is rather different from PQBE. On the other hand PQBE resembles the basic idea of the pictorial language in [7,10]. Chang's language involves three types of matching; the symbolic sub-images of Figure 15 correspond to matches of type 0, 1 and 2 of the symbolic image s. In contrast to Chang's language, PQBE involves only type 1 matching, because according to our interpretation, type 2 is considered equivalent to type 1 (they both denote the relation *NorthEast*) and type 0 yields a different set of binary direction relations (i.e., *RestrictedNorth*(M,B) instead of *NorthEast*(M,B) in symbolic image s).



M
D
B
M
D
B
D
B
M
B

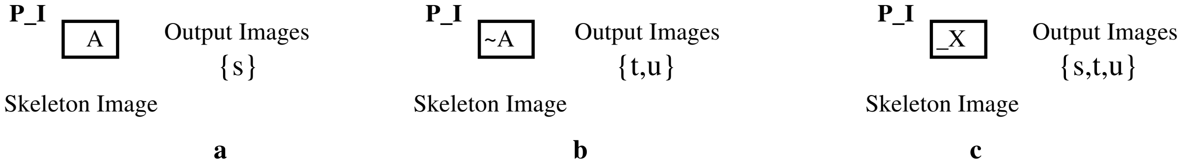
sub-image of s type 0 match type 1 match type 2 match

Fig. 15 Three types of matches

In addition to object configuration retrieval, PQBE extends Chang's pictorial language by including variable and constant reference objects, negation, union, intersection, join and a large set of additional features which we will describe in the rest of the paper.

4. Queries Involving Image Retrieval

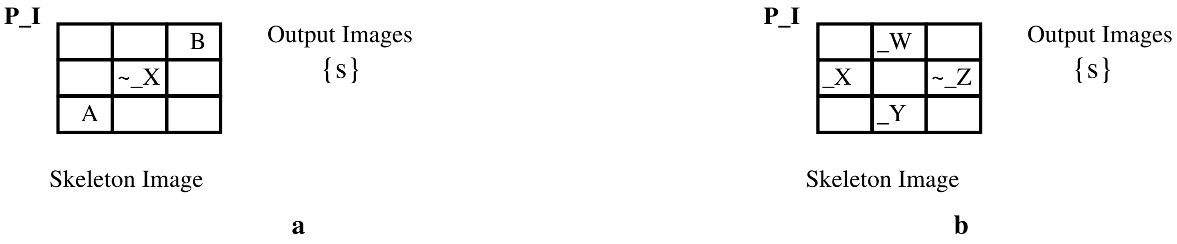
Unlike the queries of the previous sections which involve sub-image retrieval, the queries of Section 4 involve the retrieval of entire images. For instance, the first query in Figure 16 will retrieve all images that contain object A, that is, all images I such that: $A \in O(I)$. On the other hand, the query in Figure 16b will retrieve all images that do not contain object A, that is, all images I such that: $A \notin O(I)$. The third query in Figure 16 will retrieve all non-empty images in the database, i.e., all images I such that: $\exists X\ (X \in O(I))$.



P_I
A
Output Images
{s}
Skeleton Image
a
P_I
~A
Output Images
{t,u}
Skeleton Image
b
P_I
_X
Output Images
{s,t,u}
Skeleton Image
c

Fig. 16 Queries involving the retrieval of images that contain object constants or variables

We can also express queries involving the retrieval of images that contain certain configurations of objects. The first query in Figure 17 will retrieve all images in which B is *NorthEast* of A and there does not exist an object X between them; i.e., all images I such that: $\mathrm{NorthEast}(B,A) \in C(I) \wedge {\sim}\exists X\ (\mathrm{NorthEast}(X,A) \in C(I) \wedge \mathrm{SouthWest}(X,B) \in C(I))$. The second query will retrieve all images in which there exist some objects that satisfy the given configuration; i.e., all images I such that: $\exists X\ \exists Y\ \exists W\ {\sim}\exists Z\ ((\mathrm{NorthWest}(X,Y) \in C(I) \wedge \mathrm{SouthWest}(X,W) \in C(I) \wedge \mathrm{RestrictedNorth}(W,Y) \in C(I) \wedge \mathrm{NorthWest}(W,Z) \in C(I) \wedge \mathrm{RestrictedWest}(X,Z) \in C(I) \wedge \mathrm{SouthWest}(Y,Z) \in C(I))$.



P_I
B
~_X
A
Output Images
{s}
Skeleton Image
a
P_I
_W
_X
~_Z
_Y
Output Images
{s}
Skeleton Image
b

Fig. 17 Queries involving the retrieval of images that contain object configurations

Using multiple skeleton images we can express queries involving the retrieval of symbolic images that contain common objects. The query in Figure 18a, for instance, will retrieve all images which have at least a common object with image t; the result contains all images I such that: ∃X (X∈O(I) ∧ X∈O(t)). The query in Figure 18b will retrieve all images that contain an object X which is *RestrictedNorth* of A in image s (i.e., all images that contain K and F). The result consists of all images I such that: ∃X(X∈O(I) ∧ RestrictedNorth(X,A)∈C(s)). The third query will retrieve all images that contain an object X *RestrictedNorth* of another object Y, where X is *NorthWest* of B in image t and Y is *RestrictedNorth* of A in image s, i.e., ∃X∃Y (RestrictedNorth(X,Y)∈C(I) ∧ NorthWest(X,B)∈C(t) ∧ RestrictedNorth(Y,A)∈C(s)).

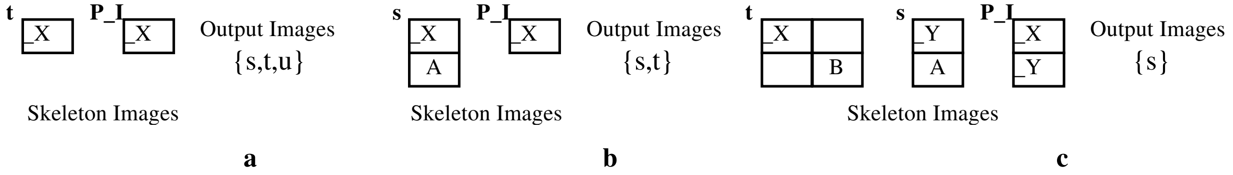

t
P_I
_X
_X
Output Images
{s,t,u}
Skeleton Images
a
s
P_I
_X
A
_X
Output Images
{s,t}
Skeleton Images
b
t
_X
B
s
_Y
A
P_I
_X
_Y
Output Images
{s}
Skeleton Images
c

Fig. 18 Queries involving common objects

PQBE can express union and intersection in image retrieval in a manner similar to sub-image (i.e., primary object) retrieval. The first query in Figure 19, for instance, will retrieve all images that either contain A, or B. This corresponds to the union operation and the result is the set of all images I that satisfy the condition: A∈O(I) ∨ B∈O(I). The query in Figure 19b will retrieve all images that contain both A and B (independently of the relation between them). This corresponds to the intersection operation and the result is the set of all images I that satisfy the condition: A∈O(I) ∧ B∈O(I). The query in Figure 19c will retrieve all images in which B is *SouthEast* of F and do not contain object C, i.e., all images I such that: C∉O(I) ∧ SouthEast(B,F)∈C(I).

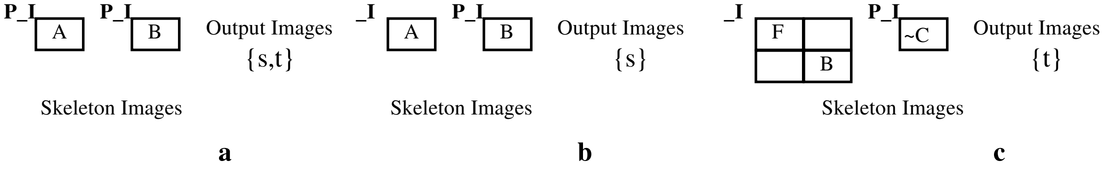

P_I
A
P_I
B
Output Images
{s,t}
Skeleton Images
a
_I
A
P_I
B
Output Images
{s}
Skeleton Images
b
_I
F
B
P_I
~C
Output Images
{t}
Skeleton Images
c

Fig. 19 Queries involving union and intersection

We can integrate the previous cases in order to express complex queries that combine union, intersection, object configurations and common objects. The query in Figure 20a, for instance, will retrieve all images that contain an object X, or an object Y, such that X is *RestrictedNorth* of Y in image s, i.e., ∃X∃Y((X∈O(I) ∨ Y∈O(I)) ∧ RestrictedNorth(X,Y)∈C(s)). The second query will retrieve all images that contain both objects X and Y such that X is *RestrictedNorth* of Y in image s, i.e., ∃X∃Y((X∈O(I) ∧ Y∈O(I)) ∧ RestrictedNorth(X,Y)∈C(s)).

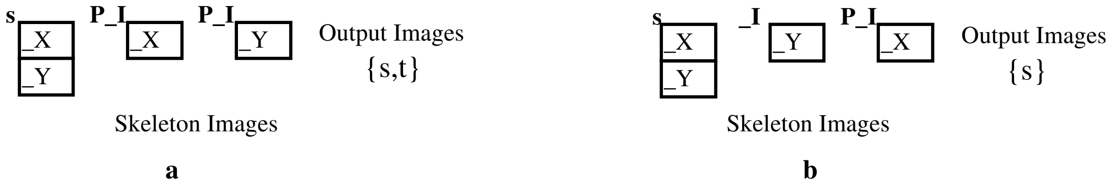


Fig. 20 Queries involving multiple features

So far PQBE retrieves spatial information explicitly stored in one or more symbolic images. The next section extends the language by allowing the retrieval of spatial information regarding objects that exist in different images. This information is not explicitly stored but it can be inferred using appropriate rules of inference.

5. Queries Involving Spatial Inference

We can use *composition* of direction relations to infer the relation between objects that exist in different symbolic images, provided that there exists a chain of common objects connecting the images. The problem of composition can be defined as "if the spatial relation between objects X and Z, and between Z and Y is known what are the possible relations between X and Y?". Frank [20] describes composition of direction relations based on the concepts of *projections* and *cone-shaped directions*. Hernandez [21] also studied composition of direction relations in 2D space and Egenhofer [22] composition of topological relations. Frank uses the notion of *Euclidean approximate* to deal with uncertainty, while in Hernandez's and Egenhofer's systems when various cases are possible the systems generate a disjunction of the potential primitive relations.

Similar to Hernandez and Egenhofer, we use disjunctions of primitive relations in order to deal with uncertainty. We define relations of lower direction accuracy as follows:

North(X,Y) ≡ NorthWest(X,Y) ∨ RestrictedNorth(X,Y) ∨ NorthEast(X,Y)
East(X,Y) ≡ NorthEast(X,Y) ∨ RestrictedEast(X,Y) ∨ SouthEast(X,Y)
South(X,Y) ≡ SouthWest(X,Y) ∨ RestrictedSouth(X,Y) ∨ SouthEast(X,Y)
West(X,Y) ≡ NorthWest(X,Y) ∨ RestrictedWest(X,Y) ∨ SouthWest(X,Y)
SameLevel(X,Y) ≡ RestrictedWest(X,Y) ∨ SamePosition(X,Y) ∨ RestrictedEast(X,Y)
SamewidtH(X,Y) ≡ RestrictedNorth(X,Y) ∨ SamePosition(X,Y) ∨ RestrictedSouth(X,Y)

For the rest of the paper we use the following abbreviations: NW for NorthWest, RN for RestrictedNorth, NE for NorthEast, RW for RestrictedWest, SP for SamePosition, RE for RestrictedEast, SW for SouthWest, RS for RestrictedSouth, SE for SouthEast, N for North, E for East, S for South, W for West, SL for SameLevel and SH for SamewidtH.

Composition tables have been commonly used to describe the results of composition. Table 1 describes all the possible compositions for the primitive direction relations. The result of composition can be a primitive relation, a low accuracy relation (N, E, S, W, SL, SH, i.e., the disjunction of three primitive relations) or the disjunction of all primitive relations (these entries of the table are denoted with symbol *T*). The table extends the composition table for *cardinal directions with neutral area* in [20] by including the relations of low accuracy. These relations are used in the cases that the symbolic images containing the initial objects are not connected through a single object, but through a chain of common objects and an intermediate composition yields a relation of lower accuracy.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		NW(Z,Y)	**RN(Z,Y)**	**NE(Z,Y)**	**RW(Z,Y)**	**SP(Z,Y)**	**RE(Z,Y)**	**SW(Z,Y)**	**RS(Z,Y)**	**SE(Z,Y)**	**N(Z,Y)**	**E(Z,Y)**	**S(Z,Y)**	**W(Z,Y)**	**SL(Z,Y)**	**SH(Z,Y)**
1	**NW(X,Z)**	NW	NW	N	NW	NW	N	W	W	T	N	T	T	W	N	W
2	**RN(X,Z)**	NW	RN	NE	NW	RN	NE	W	SH	E	N	E	T	W	N	SH
3	**NE(X,Z)**	N	NE	NE	N	NE	NE	T	E	E	N	E	T	T	N	E
4	**RW(X,Z)**	NW	NW	N	RW	RW	SL	SW	SW	S	N	T	S	W	SL	W
5	**SP(X,Z)**	NW	RN	NE	RW	SP	RE	SW	RS	SE	N	E	S	W	SL	SH
6	**RE(X,Z)**	N	NE	NE	SL	RE	RE	S	SE	SE	N	E	S	T	SL	E
7	**SW(X,Z)**	W	W	T	SW	SW	S	SW	SW	S	T	T	S	W	S	W
8	**RS(X,Z)**	W	SH	E	SW	RS	SE	SW	RS	SE	T	E	S	W	S	SH
9	**SE(X,Z)**	T	E	E	S	SE	SE	S	SE	SE	T	E	S	T	S	E
10	**N(X,Z)**	N	N	N	N	N	N	T	T	T	N	T	T	T	N	T
11	**E(X,Z)**	T	E	E	T	E	E	T	E	E	T	E	T	T	T	E
12	**S(X,Z)**	T	T	T	S	S	S	S	S	S	T	T	S	T	S	T
13	**W(X,Z)**	W	W	T	W	W	T	W	W	T	T	T	T	W	T	W
14	**SL(X,Z)**	N	N	N	SL	SL	SL	S	S	S	S	T	N	T	SL	T
15	**SH(X,Z)**	E	SH	W	E	SH	W	E	T	W	T	E	T	W	T	SH

Table. 1 Composition Table

The composition table is used by our query language in order to find the possible relations between two objects that belong to different images. The query in Figure 21, for instance, will generate a sub-image representing the relation between object A in image s and object O in image t. This sub-image is described by the following sets: $O(I)=\{A,O\}$ and $C(I) = \{R(A,O) \mid \exists W \exists R_1 \exists R_2 \ (R_1(A,W) \in C(s) \land R_2(W,O) \in C(t)) \land (R_1(X,Z) \ {*p} \ R_2(Z,Y) = R(X,Y))\}$[4]. From SouthWest(A,B)∈C(s) and SouthWest(B,O)∈C(t) and the rule: SouthWest(X,Z) *p SouthWest(Z,Y) = SouthWest(X,Y) (entry 7, 7 of the composition table) we can conclude that *SouthWest(A,O)*. The same conclusion can be reached if we use F as the common object, i.e., RestrictedSouth(A,F)∈C(s) and RestrictedWest(F,O)∈C(t) and the entry 8, 4 of the composition table. Furthermore, since composition is recursive, we can retrieve the relation between objects that exist in images that are not directly connected through a single object, but through a chain of objects. For instance, we can conclude that P is *NorthEast* of A, because it is *NorthEast* of O, which is *NorthEast* of B, which is *NorthEast* of A (using transitivity).

---

[4] The symbol $*_p$ denotes *path composition* (Frank, 1992). $R_1 (X,Z) *_p R_2 (Z,Y) = R(X,Y)$ can be substituted by an inference rule of the form $\forall X \ \forall Y \ \forall Z \ (R_1 (X,Z) \land R_2 (Z,Y)) \Rightarrow R(X,Y)$.

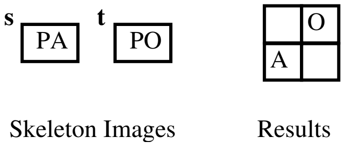


Fig. 21 Query involving the retrieval of relations between objects in different images

Since symbolic images do not preserve absolute metric information the composition relation is not always a primitive one. The query in Figure 22 will generate a sub-image I representing the relation between object C in image s and object N in image t; $O(I)=\{C,N\}$ and $C(I)=\{R(C,N) \mid \exists W \exists R_1 \exists R_2\ (R_1(C,W) \in C(s) \wedge R_2(W,N) \in C(t)) \wedge (R_1(X,Z) *p\ R_2(Z,Y) = R(X,Y))\}$. From SouthEast(C,B)∈C(s) and NorthEast(B,N)∈C(t) and the entry 9, 3 of the composition table we can conclude that *East(C,N).* Unlike the previous query where the result was one of the nine primitive relations, the relation *East* is a disjunction of three primitive relations.

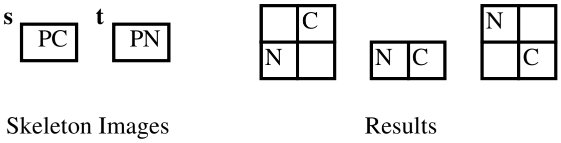


Fig. 22 Query involving the retrieval of non-primitive relation

Further refinements are possible in the presence of additional information in the form of distance relations. Table 2 illustrates the composition of primitive relations when it is known that Near(X,Z) and Far(Z,Y). As it is shown in the table, all the results of composition are primitive relations (we omit *SamePosition* because it cannot be the case SamePosition(X,Y) and Far(X,Y)). In the previous example if we also know that Near(C,B) and Far(B,N) we can conclude that NE(C,N) (entry 8,3 of Table 2). Thus only the first output array of Figure 22 is valid given the additional distance information. In the case that Near(X,Z) and Near(Z,Y), or, Far(X,Z) and Far(Z,Y) distance information is useless since it does not rule out any direction relations. Frank [23] dealt with composition of direction and distance relations at geographic scales.

		1	2	3	4	5	6	7	8
		Far(Z,Y) ∧ NW(Z,Y)	**Far(Z,Y) ∧ RN(Z,Y)**	**Far(Z,Y) ∧ NE(Z,Y)**	**Far(Z,Y) ∧ RW(Z,Y)**	**Far(Z,Y) ∧ RE(Z,Y)**	**Far(Z,Y) ∧ SW(Z,Y)**	**Far(Z,Y) ∧ RS(Z,Y)**	**Far(Z,Y) ∧ SE(Z,Y)**
1	**Near (X,Z) ∧ NW(X,Z)**	NW	NW	NE	NW	NE	SW	SW	SE
2	**Near (X,Z) ∧ RN(X,Z)**	NW	RN	NE	NW	NE	SW	RS	SE
3	**Near (X,Z) ∧ NE(X,Z)**	NW	NE	NE	NW	NE	SW	SE	SE
4	**Near (X,Z) ∧ RW(X,Z)**	NW	NW	NE	RW	RE	SW	SW	SE
5	**Near (X,Z) ∧ RE(X,Z)**	NW	NE	NE	RW	RE	SW	SE	SE
6	**Near (X,Z) ∧ SW(X,Z)**	NW	NW	NE	SW	SE	SW	SW	SE
7	**Near (X,Z) ∧ RS(X,Z)**	SW	RN	NE	SW	SE	SW	RS	SE
8	**Near (X,Z) ∧ SE(X,Z)**	NW	NE	NE	SW	SE	SW	SE	SE

Table 2 Composition table for direction and distance relations

Composition can extend the expressive power of PQBE in significant ways. For instance the first query in Figure 23a will retrieve all objects *NorthEast* of B in the database. This query, unlike the query of Figure 7a, retrieves not only the objects that co-exist in the same image with B, but also the objects that exist in other images and satisfy the spatial condition according to the composition table (i.e., object P). In general, the queries of the previous sections can be used to handle retrieval in multiple images using composition. The only difference is that in queries involving composition, the skeleton images do not correspond to particular images and therefore, they do not have superscripts denoting specific symbolic images. The query in Figure 23b, for instance, will retrieve all images that contain an object X which is *NorthEast* of B (i.e., all symbolic images in the database that contain at least one of M, E, H, D, I, O and P).

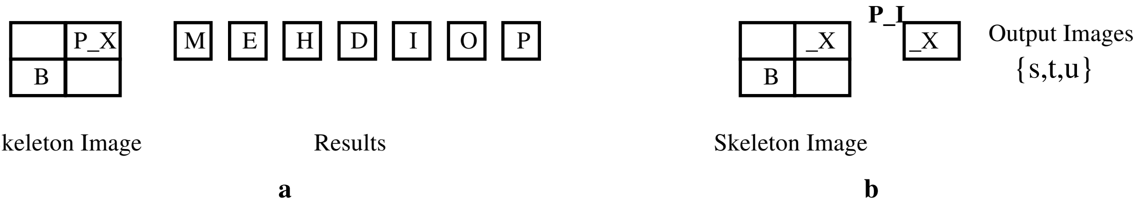
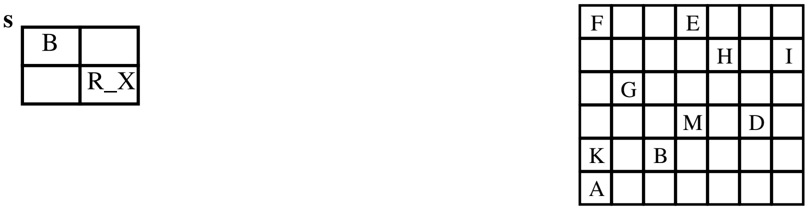


Fig. 23 Queries involving the composition table

While the previous sections refer to information retrieval, the next section is concerned with update operations. In particular we will demonstrate how PQBE can handle object removal, object insertion and other update tasks.

6. Update Operations

Any spatial knowledge representation system is expected to function in a dynamic environment in which a change in a single item of knowledge may have widespread effects. With minor extensions, PQBE can handle update operations. Object deletion from an image can be treated in a way similar to object retrieval. Instead of the character *P* that causes the contents of a variable to be printed, we will use a special character *R* that causes all the objects that satisfy a spatial query to be removed from the image array. After all the symbols that belong to the objects that satisfy the spatial conditions are deleted, the empty rows or columns are removed. For instance, the skeleton image of Figure 24a will delete from image s of Figure 1 all the objects *SouthEast* of B (objects C, L, J). The new symbolic image is illustrated in Figure 24b.



B	
	R_X

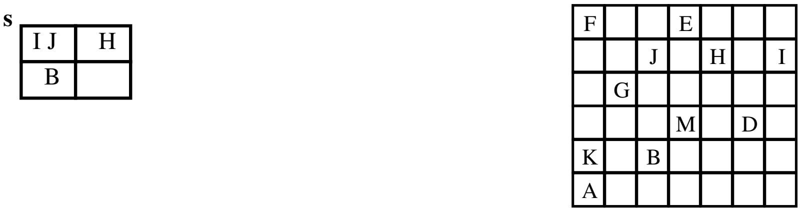
a

F			E			
				H		I
	G					
			M		D	
K		B				
A						

b

Fig. 24 Object removal using PQBE

Object insertion can also be handled by PQBE provided that the position of the object to be added is precisely specified. For example, the query in the skeleton image of Figure 25a will add an object J *RestrictedNorth* of B and *RestrictedWest* of H in the symbolic image of Figure 24b (the special character *I* denotes object insertion). The new symbolic image that results after the insertion operation is illustrated in Figure 25b.



I J	H
B	

a

F			E			
		J		H		I
	G					
			M		D	
K		B				
A						

b

Fig. 25 Object insertion using PQBE

In some cases the insertion of an object may result in several output arrays corresponding to the possible positions of the new object. For example, the insertion of another object L *RestrictedNorth* of J (without any additional restrictions) in the image of Figure 25b would result in the three symbolic images of Figure 26, since the position of L with respect to F and E is not precisely specified (the three arrays correspond to the possible composition relations between L and the objects F and E). In other cases object insertion in the presence of uncertainty may produce a very large number of output images (e.g., the insertion of an object *NorthEast* of B without additional constraints). An alternative way to insert an object in an image is by explicitly specifying the cell where the object is to be added. Furthermore, object movements within the array can be handled by removing the object from its old position and inserting it in the new one.

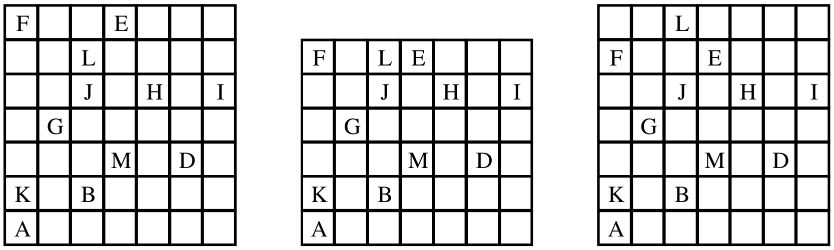
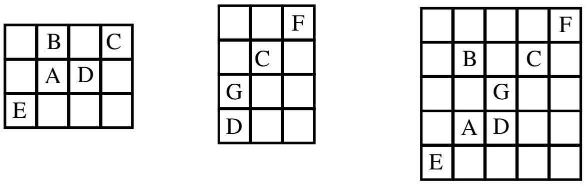


Fig. 26 Object insertion involving uncertainty

Extensions of PQBE could deal with the problems of Assimilation and Error Correction [24]. The goal of Assimilation, given an accurate representation of spatial knowledge and an accurate fact, is to augment the representation in order to include the new fact. The goal of Error Correction, given a spatial knowledge representation which is not quite accurate and a new fact which is more precise, is to improve the representation. Figure 27 illustrates how spatial knowledge existing in two images can be assimilated in a new image so that the direction relations between all pairs of objects are preserved.



old state new facts new state

Fig. 27 Example of knowledge assimilation

Both these problems can be treated by operators that take two symbolic images as arguments (i.e., one representing the initial state and one representing the new fact) and generate one or more output images that describe the possible new states.

7. Concluding Remarks

Although PQBE was first conceived as a means for retrieving information in symbolic images, we believe that it has numerous potential applications independently of the underlying implementation. For instance, all the previous queries could refer to relations among cities existing in maps of regions. Assume that a geographic application includes the map of Central European cities illustrated in Figure 28 (the projections are added to denote directions between cities).

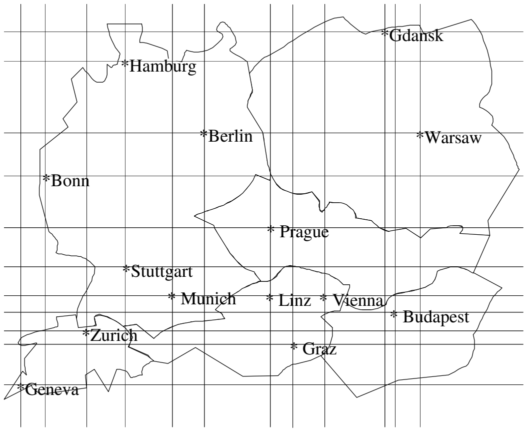


Fig. 28 Map of cities in Central Europe

The direction constraints between the cities in the map can be modelled by the symbolic image of Figure 29a, or 2D strings, spatial predicates etc. Independently of the underlying implementation, the query of Figure 29b would retrieve the cities *NorthEast* of Vienna in this map. The result contains Warsaw and Gdansk, since they are the only cities in the map constrained by the projection lines that determine north and east direction.

									Gdansk		
			Hamburg								
					Berlin						Warsaw
	Bonn										
						Prague					
			Stuttgart								
				Munich		Linz		Vienna			
										Budapest	
		Zurich									
							Graz				
Geneva											

a

Central Europe

	P_X
Vienna	

b

Fig. 29 A geographical query

By allowing more than one symbols per object in a skeleton (or symbolic) image we extend the number of direction relations that we represent. Using such skeleton images we can express topological and direction relations of high qualitative resolution (for an extensive discussion see [14]). For instance, the query in Figure 30a could be interpreted as "find all countries that have parts *North* and *NorthEast* of Austria in the map of central European Countries" (e.g., Poland). Similarly the interpretation of Figure 30b could be "find all neighbour countries *NorthWest* of Austria" (e.g., Germany). The existence of two objects in the same cell is interpreted as "neighbourhood" or "adjacency". The extension of the language to express such queries though, is not trivial and requires analysis beyond the scope of this paper. Furthermore, as in verbal

QBE, we can add *condition boxes* in order to express additional information such as distance and non-spatial conditions.

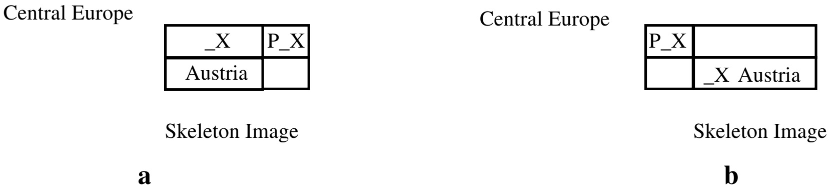
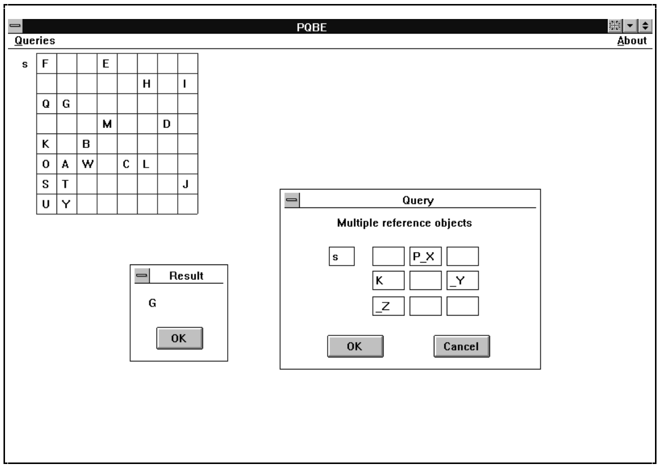

Central Europe
_X
P_X
Austria
Skeleton Image
a
Central Europe
P_X
_X Austria
Skeleton Image
b

Fig. 30 Queries involving multiple symbolic instance per object

We have implemented the larger part of PQBE using a database of symbolic images as the underlying structures. Figure 31 illustrates a skeleton image, the symbolic image to which the query refers and the result. We plan to compare PQBE with other languages that deal with space, and to use it as an interface for spatial databases independently of the underlying implementation in the future.


PQBE
Queries
About
s

F			E				
					H		I
Q	G						
			M			D	
K		B					
O	A	W		C	L		
S	T						J
U	Y						

Query
Multiple reference objects

s		P_X	
	K		_Y
	_Z		

OK
Cancel
Result

G

OK

Fig. 31 An implementation

In comparison to verbal QBE and other (spatial) query languages, PQBE provides a more intuitive and easy to use interface because of its close correspondence with the structure of 2D space. Unlike verbal languages it does not presume knowledge of keywords for spatial relations on behalf of the user, nor familiarity with database languages but complex spatial conditions are expressed in a straightforward manner. Furthermore, several features can be incorporated to increase the expressive power of PQBE and make it friendlier to the user.

Acknowledgements

This research has been supported by the Esprit Basic Research Program 6881 (**A**lgorithms, **M**odels, **U**ser and **S**ervice **In**terfaces for **G**eography). Dimitris Papadias has also been supported by NSF - IRI 9221276. Part of this work was done while he was with the Department of

Geoinformation, Technical University of Vienna. The authors would like to thank Max Egenhofer; his comments contributed significantly to the improvement of the paper.

References

1. D. Mark (1992) Counter-Intuitive Geographic "Facts": Clues for Spatial Reasoning at Geographic Scales. In Frank, A., Campari, I., Formentini, U. (eds.) *International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Pisa, Italy. Springer Verlag.

2. D. Papadias, A. Frank & M. Koubarakis (1994) Constraint-Based Reasoning in Geographic Databases: The Case of Symbolic Arrays. *2nd ICLP Workshop on Deductive Databases*, Santa Margherita Ligure, Italy. Publication Series of the GMD Society.

3. P. Sistla, C. Yu & R. Haddad (1994) Reasoning about Spatial Relationships in Picture Retrieval Systems. *20th Very Large Data Bases Conference*.

4. D. Papadias, Y. Theodoridis, & T. Sellis (1994) The Retrieval of Direction Relations Using R trees. *5th Conference on Database and Expert Systems Applications*, Athens. Springer Verlag LNCS.

5. R. Jackendoff (1983) *Semantics and Cognition*. MIT Press, Cambridge, MA.

6. A. Herskovits (1986) *Language and Spatial Cognition*. Cambridge University Press, Cambridge, UK.

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

8. J. Glasgow & D. Papadias (1992) Computational Imagery. *Cognitive Science*, 16, pp. 355-394.

9. D. Papadias & T. Sellis (1993) The Semantics of Relations in 2D Space Using Representative Points: Spatial Indexes. In Frank, A., Campari, I. (eds.) *European Conference on Spatial Information Theory*, Elba, Italy. Springer Verlag LNCS.

10. S.K. Chang, C.W. Yan, T. Arndt & D. Dimitroff (1988) An Intelligent Image Database System. *IEEE Transactions on Software Engineering, Special issue on Image Databases*, pp. 681-688.

11. J. Glasgow (1994) Spatial Reasoning in Indeterminate Worlds. *In the Proceedings of AAAI Conference.*

12. P.D. Holmes & E. Jungert (1992) Symbolic and Geometric Connectivity Graph Methods for Route Planning In Digitized Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, pp. 549-565.

13. S-Y. Lee, M-C. Yang & J-W. Chen (1992) Signature File as a Spatial Filter for Iconic Image Database. *Journal of Visual Languages and Computing*, vol. 3, pp. 373-397.

14. D. Papadias & T. Sellis (1994) The Qualitative Representation of Spatial Knowledge in two-dimensional Space. *Very Large Data Bases Journal, Special Issue on Spatial Databases*, Vol. 4, pp. 100-138

15. M. Zloof (1977) Query-by-Example: A Data Base Language. *IBM Systems Journal*, No 4, pp. 324-343.

16. N. Roussopoulos, C. Faloutsos & T. Sellis (1988) An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering, Special issue on Image Databases*, pp. 639-650.

17. M. Egenhofer (1994) Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, Vol 6, pp. 86-95.

18. D. Papadias & T. Sellis (1994) A Pictorial Language for the Retrieval of Spatial Relations from Image Databases. *6th International Symposium on Spatial Data Handling*. Taylor Francis.

19. N.S. Chang & K.S. Fu (1980) Query-by-Pictorial Example. *IEEE Transactions on Software Engineering*, Vol 6, pp. 519-524.

20. A. Frank Qualitative Spatial Reasoning: Cardinal Directions as an Example. To appear in the *International Journal of Geographic Information Systems.*

21. D. Hernandez (1994) *Qualitative Representation of Spatial Knowledge*. Springer Verlag LNAI.

22. M. Egenhofer (1991) Reasoning about Binary Topological Relations. In Günther, O. and Schek, H.J. (eds.) *Advances in Spatial Databases. Second Symposium*. Springer Verlag LNCS.

23. A. Frank (1992) Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing*, 3, pp. 343-371.

24. E. Davis (1986) *Representing and Acquiring Geographic Knowledge*. Morgan Kaufmann, Los Altos, CA.