Object Localization Based on Directional Information: Case of 2D Vector Data

Stelian Coros JingBo Ni Pascal Matsakis

CIS Department, University of Guelph
Guelph, ON N1G 2W1, Canada
+1 519 824 4120

scoros@gmail.com jni@uoguelph.ca matsakis@cis.uoguelph.ca

ABSTRACT

If you were told that some object A was perfectly (or somewhat, or not at all) in some direction δ (e.g., west, above-right) of some reference object B, where in space would you look for A? Cognitive experiments suggest that you would mentally build a spatial template. Using essentially angular deviation, you would partition the space into regions where the relationship "in direction δ of B" holds (to various extents) and regions where it does not hold. You would then be able to locate the objects for which the relationship holds best, and find A. Spatial templates, therefore, represent directional spatial relationships to reference objects (e.g., "east of the post office"). Note that other names can also be found in the literature (e.g., fuzzy landscape, applicability structure, potential field). There exists a very simple and yet cognitively plausible way to mathematically model a spatial template without sacrificing the geometry of the reference object (i.e., the object is not approximated through its centroid or minimum bounding rectangle). In case of 2D raster data, exact calculation of the model can easily be achieved but is computationally expensive, and tractable approximation algorithms were proposed. In case of 2D vector data, exact calculation of the model is not conceivable. In previous work, we introduced the concept of the F-template. We discussed the case of 2D raster data and designed, based on this concept, an efficient approximation algorithm for spatial template computation. The algorithm is faster, gives better results, and is more flexible than its competitors. Here, comparable advances are presented in the case of 2D vector data. These advances are of particular interest for spatial query processing in Geographic Information Systems.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *representations.* I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement – *feature representation.* I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *object recognition.* I.5.1 [**Pattern Recognition**]: Models – *fuzzy set, geometric.* I.6.5 [**Simulation and Modeling**]: Model development. H.2.8 [**Database Management**]: Database Applications – *spatial databases and GIS.*

General Terms

Algorithms

Keywords

Spatial relationships, directional relationships, spatial templates, directional templates, F-templates

1. INTRODUCTION

Space plays a fundamental role in human cognition. In everyday situations, it is often viewed as a construct induced by spatial relationships, rather than as a container that exists independently of the objects located in it. Spatial relationships, therefore, have been thoroughly investigated in many disciplines, including cognitive science, psychology, linguistics, geography and artificial intelligence. They act as a connecting link between visually perceived data and natural language, and an important part of research naturally deals with two types of tasks: those related to the translation of visual information into linguistic expressions (e.g., automatic digital image analysis and description), and those related to the translation of linguistic expressions into visual information (e.g., spatial query processing in Geographic Information Systems). In this paper, we focus on directional (also called projective) relationships (e.g., front, south, above). The past ten to fifteen years have seen significant advancements in the development of mathematical and computational models of these relationships [1], [2], [3], [4], [5], [6], [7], [8], [9]. Tasks of the first type require from such models the capability to identify which relationships hold best between any two objects. Tasks of the second type require different capabilities. Given a directional relationship to a reference object (e.g., "east of the post office"), the models should be able to identify the objects for which the relationship holds best, and also to distinguish regions where it holds from regions where it does not hold. These regions, of course, blend into one another. Different names can be found in the literature (e.g., "fuzzy landscape" [10], "spatial template" [11], "applicability structure" [12], "potential field" [13]). Here, we will use the term "directional spatial template" (or "template",

ACM-GIS'06, November 10–11, 2006, Arlington, Virginia, USA.

for short). The models mentioned earlier (references [1] to [9]) address tasks of the first type, or approximate the reference object through its minimum bounding rectangle, or are only able to distinguish a very limited number of regions. There exists, however, a simple and yet cognitively plausible way to model a template without sacrificing the geometry of the objects or restricting the number of regions. In case of 2D raster data, exact calculation of the model can easily be achieved but is computationally expensive, and tractable approximation algorithms were proposed. The issue was discussed in [14]. In case of 2D vector data, exact calculation of the model is not conceivable. A new, efficient and flexible approximation algorithm is introduced in Section 3. An experimental study, presented in Section 4, validates the approach. Conclusions and directions of future work are given in Section 5. First, in Section 2, we introduce some notations and briefly review two important concepts.

2. BASIC AND F-TEMPLATES

2.1 Notations

In the following, $\Re$ denotes the set of real numbers and $\mathcal{P}$ the Cartesian plane. μ is a continuous mapping from $\Re$ into [0,1], periodic with period 2π, even, non increasing on $[0,\pi]$, and such that $\mu(0)=1$ and $\mu(\pi/2)=0$ (Fig. 1).

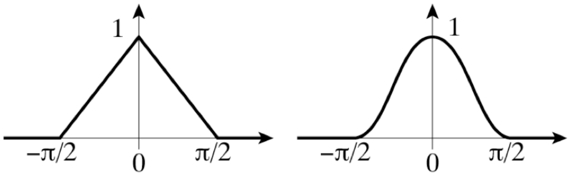

Figure 1. Two possible functions μ.

An *object* is a non-empty subset of $\mathcal{P}$. For any two points q and p in $\mathcal{P}$, with $q \neq p$, the expression $\angle(q,p)$ represents the direction of the vector $\overrightarrow{qp}$. It is a value that belongs to the interval $[-\pi,\pi)$. Directional spatial relationships defy precise definitions, and Freeman proposed that fuzzy set theory should be applied [15]. The idea has been widely accepted. For any direction δ and any two objects A and B, the expression $d(\delta,A,B)$ represents the degree of truth of the proposition "A is in direction δ of B". It is a value that belongs to [0,1]. For instance, $d(0,A,B)=1$ might express the fact that the object A is perfectly to the right of B, the equality $d(-\pi/2,A,B)=0.5$ that A is somewhat behind B, and $d(\pi/2,A,B)=0$ that A is not at all to the north of B (Fig. 2).

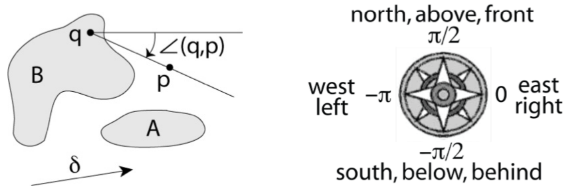

Figure 2. Points, objects and directions. Is A in direction δ of B?

2.2 Basic Templates

Let A and B be two objects. If you were told that A was perfectly (or somewhat, or not at all) in direction δ (e.g., west, above-right) of B, where in space would you look for A? Cognitive experiments suggest that you would mentally build a spatial template [11], [16], [17]. Using essentially angular deviation, you would partition the space into regions where the relationship "in direction δ of B" holds to various degrees. It therefore makes sense to model this template by a mapping from $\mathcal{P}$ into $\Re$, such as $S^{\delta B}$.

$$\forall p \in \mathcal{P},\ S^{\delta B}(p) = \sup_{q \in B-\{p\}} \mu(\angle(q,p)-\delta) \qquad (1)$$

$S^{\delta B}$ is the *basic directional spatial template* induced by B in direction δ. As an example, Fig. 3b shows the basic template induced by some reference building in direction north. The brighter the area, the higher the value $S^{\delta B}(p)$, i.e., the more it is considered that the area is north of the reference building. Note that $S^{\delta B}$ allows the proposition "A is in direction δ of B" to be readily assessed for any object A. The degree of truth $d(\delta,A,B)$ of this proposition can be set to, e.g., $\sup_{p\in A} S^{\delta B}(p)$, or $\inf_{p\in A} S^{\delta B}(p)$. These two values correspond to the most optimistic and most pessimistic points of view (Fig. 3c). They can also be interpreted as a possibility degree and a necessity degree [10].

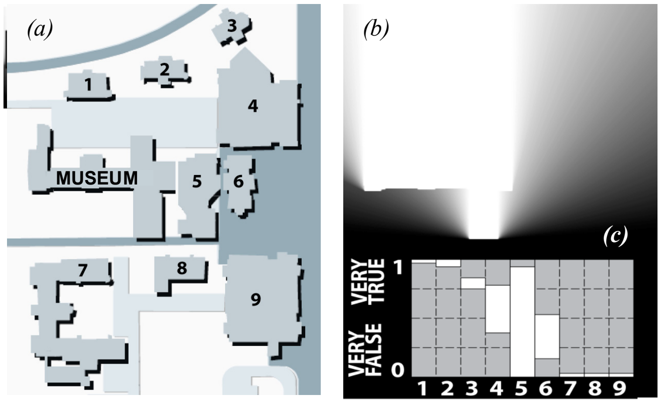

Figure 3. *(a)* Neighborhood map. *(b)* Basic spatial template: show me where the north is (relative to the museum). μ is as in Fig. 1a. *(c)* Are the buildings 1 to 9 north of the museum? The white bars represent all possible points of view.

In case of raster data, the algorithm that corresponds to Eq. 1 is straightforward. Computationally, however, it is prohibitively expensive, and tractable approximation algorithms can be found in the literature. The issue is discussed in [14]. In case of vector data (i.e., each object is defined by a polygon, or by a finite set of polygons), $S^{\delta B}(p)$ can easily be computed for any individual point p of the Cartesian plane (in the right-hand side of Eq. 1, replace sup with max and B with its set of vertices). Practically, however, $S^{\delta B}(p)$ cannot be computed for *every* point p, and the basic template $S^{\delta B}$ needs to be approximated in an efficient and appropriate way. To our knowledge, this issue has not been examined yet.

2.3 F-Templates

Basic directional spatial templates can be seen as particular F-templates. The latter were introduced in [14] as a concept dual to that of the F-histogram (described much earlier, in [18] [4]). Let δ be a direction and B an object. For any θ in $[0,\pi)$ and any p in $\mathcal{P}$, let $B_p(\theta)$ denote the intersection of B with the line that runs in direction θ and passes through p. This intersection is a *longitudinal section* of B (or *section*, for short). See Fig. 4. The set of all possible longitudinal sections is denoted by $\mathcal{L}$. Consider some section J, i.e., some element J of $\mathcal{L}$. There exists a line Δ_J that includes J. This line runs in some direction $\theta_J \in [0,\pi)$. Assume $p \in \Delta_J$. As illustrated by Fig. 5, the symbols $\Delta^-_{J,p}$ and $\Delta^+_{J,p}$ denote the two half-lines such that

$$\Delta^-_{J,p} \cup \Delta^+_{J,p} = \Delta_J \text{ and } \Delta^-_{J,p} \cap \Delta^+_{J,p} = \{p\} \tag{2}$$

$$\forall\, q \in \Delta^-_{J,p} - \{p\}, \quad \angle(q,p) = \theta_J - \pi \tag{3}$$

$$\forall\, q \in \Delta^+_{J,p} - \{p\}, \quad \angle(q,p) = \theta_J \tag{4}$$

An *F-template* induced by B in direction δ is a mapping $F^{\delta B}$ from $\mathcal{P}$ into $\Re$. In "$F^{\delta B}$", the letter "F" denotes a function from $\mathcal{P} \times \Re \times \mathcal{L}$ into $\Re$. (The letter "F" in the expression "F-template", however, is not dissociable from the word "template", and does not refer to any specific function.) The value $F^{\delta B}(p)$ is a combination of the $F(p,\delta,B_p(\theta))$ values, for all θ (Fig. 4). In the rest of this paper

$$\forall\, p \in \mathcal{P}, \quad F^{\delta B}(p) = \sup_{\theta\in[0,\pi)} F(p,\delta,B_p(\theta)) \tag{5}$$

and the function F is defined as follows, for any crisp section J, direction δ, and point p aligned with J.

$$\text{If } J=\varnothing \text{ or } J=\{p\} \text{ then } F(p,\delta,J)=0, \tag{6}$$

$$\text{else if } J \subset \Delta^-_{J,p} \text{ then } F(p,\delta,J) = \mu((\theta_J - \pi) - \delta), \tag{7}$$

$$\text{else if } J \subset \Delta^+_{J,p} \text{ then } F(p,\delta,J) = \mu(\theta_J - \delta), \tag{8}$$

$$\text{else } F(p,\delta,J) = \max\{\, \mu((\theta_J - \pi) - \delta)\,,\, \mu(\theta_J - \delta)\,\}. \tag{9}$$

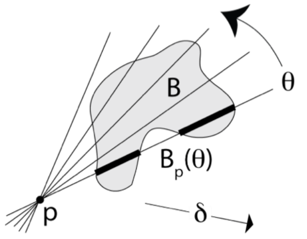

Figure 4. Longitudinal sections and F-templates. Here, the longitudinal section $B_p(\theta)$ is the union of two segments. Each line gives an $F(p,\delta,B_p(\theta))$ value. $F^{\delta B}(p)$ is a combination of the $F(p,\delta,B_p(\theta))$ values, for all θ.

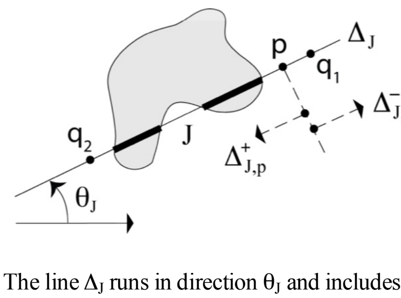

Figure 5. The line Δ_J runs in direction θ_J and includes the section J. The point p splits Δ_J into two half-lines, $\Delta^-_{J,p}$ and $\Delta^+_{J,p}$. The point q_1 in $\Delta^-_{J,p}$ is such that $\angle(q_1,p) = \theta_J - \pi$. The point q_2 in $\Delta^+_{J,p}$ is such that $\angle(q_2,p) = \theta_J$.

It is easy to show that the corresponding F-template $F^{\delta B}$ is equal to the basic directional spatial template $S^{\delta B}$ presented in Section 2.2. We have $F^{\delta B} = S^{\delta B}$. However, in case of raster data, Eqs. 5 to 9 lead to an efficient approximation algorithm, which is faster, gives better results, and is more flexible than its competitors [14]. The case of vector data is discussed in Section 3.

3. CASE OF 2D VECTOR DATA

3.1 Principle

From now on, we assume that each object is defined by a polygon. As mentioned in Section 2.2, the basic directional spatial template $S^{\delta B}$ needs to be approximated in an efficient and appropriate way. The solution proposed here is based on the concept of the F-template. As illustrated by Fig. 6, the plane is partitioned into a set $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ of regions, where N is an integer greater than 1,

$$\mathcal{R}_0 = \{p \in \mathcal{P} \mid F^{\delta B}(p) \le \mu_1\}, \tag{10}$$

$$\mathcal{R}_1 = \{p \in \mathcal{P} \mid \mu_1 < F^{\delta B}(p) < \mu_2\}, \tag{11}$$

$$\forall i \in 2..N-1,\ \mathcal{R}_i = \{p \in \mathcal{P} \mid \mu_i \le F^{\delta B}(p) < \mu_{i+1}\}, \tag{12}$$

$$\mathcal{R}_N = \{p \in \mathcal{P} \mid \mu_N \le F^{\delta B}(p)\}, \tag{13}$$

and $\mu_1 = 0 < \mu_2 < \ldots < \mu_N \le \mu_{N+1} = 1$. Clearly, the partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ represents an approximation of the template $F^{\delta B}$ (i.e., $S^{\delta B}$). The regions $\mathcal{R}_i$ are not determined directly. Instead, other regions R_i are calculated:

$$R_1 = \{p \in \mathcal{P} \mid \mu_1 < F^{\delta B}(p)\} \tag{14}$$

$$\text{and} \quad \forall i \in 2..N,\ R_i = \{p \in \mathcal{P} \mid \mu_i \le F^{\delta B}(p)\}. \tag{15}$$

We have $\mathcal{R}_0 = \mathcal{P} - R_1$, $\mathcal{R}_1 = R_1 - R_2$, $\mathcal{R}_2 = R_2 - R_3$, … $\mathcal{R}_{N-1} = R_{N-1} - R_N$ and $\mathcal{R}_N = R_N$. The R_i are nested, i.e., $R_1 \supset R_2 \supset \ldots \supset R_N$. Each R_i is an *unbounded polygon*, i.e., its boundary is the union of n−1 edges, $(r_1r_2]$, $[r_2r_3]$, $[r_3r_4]$, … $[r_{n-2}r_{n-1}]$ and $[r_{n-1}r_n)$, where n is an integer greater than 2. These edges are line segments, except $(r_1r_2]$ and $[r_{n-1}r_n)$, which are half-lines. Consider the angle

$$\theta_i = \inf\{\theta \in [0,\pi/2] \mid \mu(\theta) = \mu_1\} \text{ if } i=1, \tag{16}$$

$$\theta_i = \sup\{\theta \in [0,\pi/2] \mid \mu(\theta) = \mu_i\} \text{ if } i \in 2..N. \tag{17}$$

The half-line $(r_1r_2]$ originates from r_2, runs in direction $\delta+\theta_i$ and passes through r_1. Similarly, $[r_{n-1}r_n)$ originates from r_{n-1}, runs in

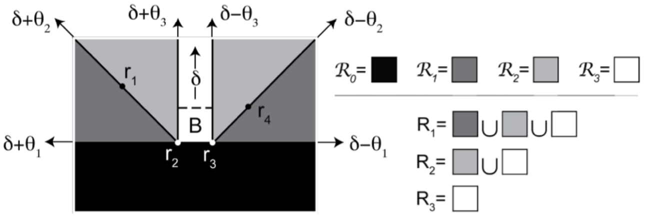

Figure 6. An approximation of the template $F^{\delta B}$. We have: $\forall p \in \mathcal{R}_0$, $F^{\delta B}(p)=0$ and $\forall p \in \mathcal{R}_1$, $0<F^{\delta B}(p)<0.5$ and $\forall p \in \mathcal{R}_2$, $0.5 \le F^{\delta B}(p) < 1$ and $\forall p \in \mathcal{R}_3$, $F^{\delta B}(p)=1$. The partition $\{\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ is defined through the nested unbounded polygons R_1, R_2 and R_3. The boundary of R_2 is $(r_1r_2] \cup [r_2r_3] \cup [r_3r_4)$. Here, μ is as in Fig. 1a, N=3, $\mu_1=0$, $\mu_2=0.5$, $\mu_3=1$, $\theta_1=\pi/2$, $\theta_2=\pi/4$, $\theta_3=0$.

direction $\delta-\theta_i$ and passes through r_n. The region R_i can actually be seen as the union of an infinite number of half-lines: each one originates from a boundary point and runs in a direction comprised between $\delta-\theta_i$ and $\delta+\theta_i$. The approach and above facts concerning R_i derive from Eqs. 5 to 9. The idea is to focus on 1D entities (lines) and not, as in Eq. 1, on 0D entities (points). Section 3.2 describes how R_i is calculated. Section 3.3 deals with the assessment of the proposition "A is in direction δ of B", for any object A. Finally, Section 3.4 shows how the partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ can be visualized.

3.2 Template Calculation

As seen in Section 3.1, an approximation of the directional spatial template $F^{\delta B}$ (i.e., $S^{\delta B}$) can be represented by a partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ of the Cartesian plane, where each region $\mathcal{R}_i$ corresponds to a range of template values (Eqs. 10 to 13). This partition is actually defined through nested unbounded polygons R_i (Eqs. 14 and 15). In this section, we describe the algorithm for the calculation of R_i. It is similar to silhouette determination algorithms developed for computer graphics applications. The idea behind it is to imagine that B is illuminated by a light source infinitely far away. The light source is moving, and the direction of the rays varies from $\delta+\theta_i$ to $\delta-\theta_i$, where θ_i is the angle associated with R_i (Eqs. 16 and 17). A point p belongs to R_i if it belongs to B or is in the shadow of B at some moment.

The reference object B is represented by a list $(b_1, b_2, \ldots b_m)$ of vertices, and the region R_i by $(r_1, r_2, \ldots r_n)$ (Section 3.1). The boundaries are traversed counterclockwise. Note that $(b_1, b_2, \ldots b_m)$, $(b_2, b_3, \ldots b_m, b_1)$, … $(b_m, b_1, b_2, \ldots b_{m-1})$ represent the same (bounded) polygon, while $(r_1, r_2, \ldots r_n)$, $(r_2, r_3, \ldots r_n, r_1)$, … $(r_n, r_1, r_2, \ldots r_{n-1})$ represent different unbounded polygons. Consider Fig. 7a. The edges facing the light source are $[b_2b_3]$, $[b_3b_4]$, $[b_5b_6]$, $[b_6b_7]$, $[b_7b_8]$ and $[b_8b_9]$ (because of obstructions, however, $[b_3b_4]$ and part of $[b_2b_3]$ are in the dark). A ray could be shot from b_1, b_2, b_5, b_9 or b_{10} in the same direction as the rays from the light source. Among these five vertices, however, only b_2, b_5 and b_9 connect edges that face the light source to edges that do not face it. b_5 is called a *key vertex*: a ray shot from b_5 would intersect B (in b). The other vertices, b_2 and b_9, are *stop vertices*: a ray shot from b_2 or b_9 would not intersect B. The algorithm includes four steps, which are described below and illustrated by Figs. 7bcde. When a key vertex b_k is encountered, B is modified as follows. Let b be the point where a ray shot from b_k would intersect B. Let e be the edge containing b_k and not facing the light source. b is added to the list of vertices and all vertices between b_k and b, via e, are removed.

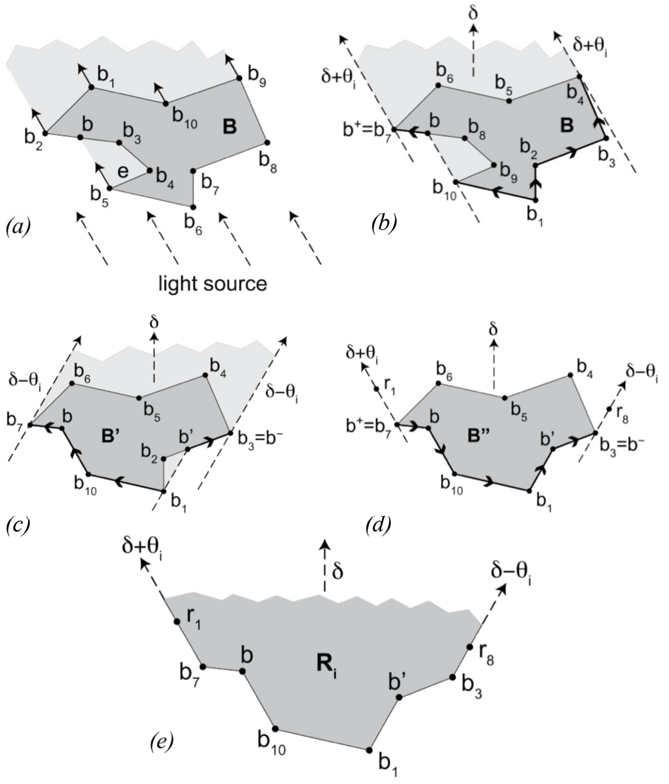

Figure 7. Calculation of the unbounded polygon R_i.
(a) The object B, and the idea behind the algorithm.
(b) At the end of step 0. *(c)* At the end of step 1.
(d) At the end of step 2. *(e)* At the end of step 3.

Step 0

The list $(b_1, b_2, \ldots b_m)$ of vertices is rearranged such that b_1 is the first vertex encountered in direction δ (compare Figs. 7a and 7b).

Step 1 (Fig. 7b)

The rays are in direction $\delta+\theta_i$, where θ_i is the angle associated with R_i (Eqs. 16 and 17). The boundary of B is traversed counterclockwise, starting from b_1, and every key vertex triggers a modification of B as described above. The traversal ends when a stop vertex is encountered. The boundary is then traversed clockwise. Again, every key vertex triggers a modification of the object, and the traversal ends when a stop vertex b^+ is encountered. At the end of this step, B has been transformed into B'.

Step 2 (Fig. 7c)

The rays are now in direction $\delta-\theta_i$. The boundary of B' is traversed *clockwise*, starting from b_1, and every key vertex triggers a modification of B'. The traversal ends when a stop vertex is encountered. The boundary is then traversed counterclockwise. Every key vertex triggers a new modification, and the traversal ends when a stop vertex b^- is encountered. At the end of this step, B' has been transformed into B".

Step 3 (Fig. 7d)

The boundary of B" is traversed counterclockwise from b^+ to b^-. The vertices $r_2= b^+$, r_3, … $r_{n-1}= b^-$ are successively encountered. Let r_1 be a point such that the half-line $(r_1r_2]$ runs in direction $\delta+\theta_i$ and let r_n be a point such that $[r_{n-1}r_n)$ runs in direction $\delta-\theta_i$. The unbounded polygon R_i is represented by $(r_1, r_2, \ldots r_n)$ (as in Fig. 7e).

If the reference object B is convex, the number of key vertices is 0, and the algorithm runs in $\mathcal{O}(m)$ time. If B is concave, the number of key vertices might be of order m. For each key vertex, some intersection point b must be computed (as in Fig. 7a). The number of edges to be considered in order to find b might be of order m. Hence, the algorithm runs in $\mathcal{O}(m^2)$ time, worst case scenario.

3.3 Assessment of the Proposition "A is in direction δ of B"

As seen in Section 3.1, an approximation of the directional spatial template $F^{\delta B}$ can be represented by a partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ of the Cartesian plane, where the region $\mathcal{R}_0$ corresponds to the template value 0, and each other region $\mathcal{R}_i$ corresponds to the range of template values between μ_i and μ_{i+1} (Eqs. 10 to 13). Given this approximation, the proposition "A is in direction δ of B" can easily be assessed for any object A. Its degree of truth d(δ,A,B) can be set to, e.g.,

$$d_{min} = \min_{i\in 1..N \mid A\cap\mathcal{R}_i \neq \varnothing} \mu_i \,, \tag{18}$$

$$d_{max} = \max_{i\in 1..N \mid A\cap\mathcal{R}_i \neq \varnothing} \mu_{i+1} \tag{19}$$

or

$$d_{ave} = \Sigma_{i\in 1..N \mid A\cap\mathcal{R}_i \neq \varnothing} \frac{\text{area}(A\cap\mathcal{R}_i)}{\text{area}(A)} \frac{\mu_i + \mu_{i+1}}{2} \,. \tag{20}$$

If $A\cap\mathcal{R}_i = \varnothing$ for all i in 1..N, i.e., if $A\subset\mathcal{R}_0$, then it is assumed that $d_{min} = d_{max} = d_{ave} = 0$. d_{min} is an approximation of $\inf_{p\in A} F^{\delta B}(p) = \inf_{p\in A} S^{\delta B}(p) = \inf_{p\in A} \sup_{q\in B-\{p\}} \mu(\angle(q,p)-\delta)$. It corresponds to the most pessimistic point of view. d_{max} is an approximation of $\sup_{p\in A} F^{\delta B}(p) = \sup_{p\in A} S^{\delta B}(p) = \sup_{p\in A} \sup_{q\in B-\{p\}} \mu(\angle(q,p)-\delta)$ and corresponds to the most optimistic point of view. See Section 2.2. Finally, d_{ave} (ave as in average) might be seen as a more reasonable point of view.

The regions $\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N$, however, are only known through the unbounded polygons R_i (Section 3.2). Here is how to calculate d_{min}, d_{max} and d_{ave}:

```
$d_{min} \leftarrow 1$
$d_{max} \leftarrow 0$
$d_{ave} \leftarrow 0$
FOR i from 1 to N DO
    IF $A \cap R_i$ is not empty THEN
        $d_{min} \leftarrow \min\{d_{min}, \mu_i\}$
        $d_{max} \leftarrow \max\{d_{max}, \mu_{i+1}\}$
        $d_{ave} \leftarrow d_{ave} + \text{area}(A\cap R_i)*\mu_{i+1}$
IF $d_{max}$ is 0 THEN $d_{min} \leftarrow 0$
ELSE $d_{ave} \leftarrow d_{ave} / (2*\text{area}(A))$
```

3.4 Template Visualization

Here again, an approximation of $F^{\delta B}$ is represented by a partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$, where the region $\mathcal{R}_0$ corresponds to the template value 0, and where each other region $\mathcal{R}_i$ corresponds to the range of template values between μ_i and μ_{i+1} (Section 3.1). This approximation can easily be visualized through an 8-bit gray level image. The gray level value assigned to $\mathcal{R}_0$ is 0 and the gray level value assigned to each other $\mathcal{R}_i$ is the closest integer to $255(\mu_i+\mu_{i+1})/2$.

The regions $\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N$, however, are only known through the unbounded polygons R_i (Section 3.2). Here is how to obtain the display image. First, all gray level values are set to 0 (the display image is black). Then, $R_1, R_2, \ldots R_N$ are painted in this order, one after the other, and on top of each other. For all i in 1..N, the gray level value assigned to R_i is the closest integer to $255(\mu_i+\mu_{i+1})/2$.

4. Experimental Study

Three sets of experiments are performed. First, in Section 4.1, specific directional spatial templates are calculated and used for the assessment of specific propositions (like "the post office is north of the museum"). Then, in Section 4.2, the quality of the templates—which depends on the number N of template regions—is examined. Finally, Section 4.3 deals with computational efficiency, which depends both on N and the number m of object vertices. For all experiments, the function μ is as in Fig. 1a. Moreover: $\forall i\in 1..N$, $\mu_i = (i-1)/(N-1)$.

4.1 Are these Buildings North of the Museum?

Here, the direction δ is π/2 (north) and the reference object B is the museum in the neighborhood map of Fig. 3a. The template $F^{\delta B}$ is approximated as in Sections 3.1 and 3.2, by a partition $\{\mathcal{R}_0, \mathcal{R}_1, \ldots \mathcal{R}_N\}$ of the Cartesian plane. Consider, for each i in 1..9, the following proposition: "the building i is north of the museum". Its degree of truth is set to d_{min} (pessimistic point of view), d_{max} (optimistic point of view) and then d_{ave} (moderate). See Section 3.3. These 9×3=27 values are calculated for different template approximations, i.e., N=6, 18, 130. The results, shown in Fig. 8, are consistent, even when the template is represented by a relatively small number of regions.

The case of raster data is also considered. The building i (object A) and the museum (object B) are then sets of pixels in a 607×757 digital image. As mentioned in Section 3.3, the value d_{min} is an approximation of $\inf_{p\in A} \sup_{q\in B-\{p\}} \mu(\angle(q,p)-\delta)$. In the image domain, this expression becomes $d_{min} = \min_{P\in A} \max_{Q\in B-\{P\}} \mu(\angle(Q,P)-\delta)$, where P and Q denote pixels. The values d_{max} and d_{ave} can be assessed using similar formulas. (Such computations are, of course, very time consuming. See [14] for less expensive methods.) Figure 8 shows that the results are consistent, whether the objects are in raster form or in vector form.

4.2 Quality Analysis

Quality is measured through error curves and error difference images (Figs. 9 and 10). Assume an approximation of some template $F^{\delta B}$ is calculated (Sections 3.1 and 3.2) and then visualized through a 256×256 digital image (Section 3.4). Each pixel P corresponds to some point p of the Cartesian plane, and the gray level of P corresponds (up to the multiplicative constant 255) to an approximation of the template value $F^{\delta B}(p)$. However, as mentioned earlier (Section 2.2), the exact template value—and hence the absolute difference between the two—can also be computed. The sum of absolute differences, for all P in the display image, is a measure, SAD, of the quality of the approximation.

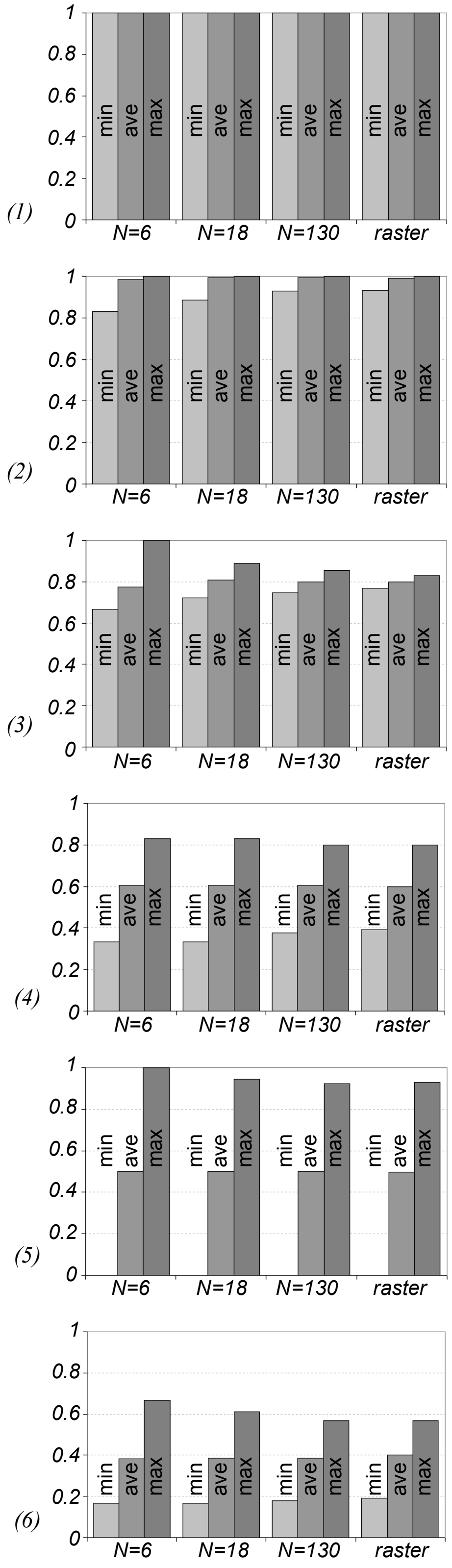

Figure 8. Are the buildings 1 to 6 north of the museum (Fig. 3a)? All possible points of view. For the buildings 7 to 9, the answer is definitely no ($d_{min} = d_{max} = d_{ave} = 0$).

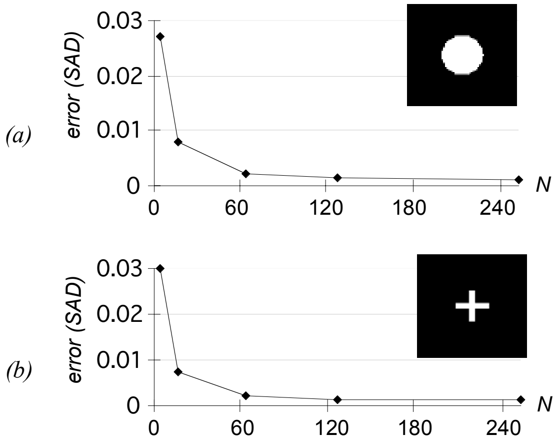

Figure 9. Error curves. The reference object B is as shown, *(a)* convex, *(b)* concave, and the direction δ is π/4 (northeast). The higher the number of regions, the more accurate the approximation of the template.

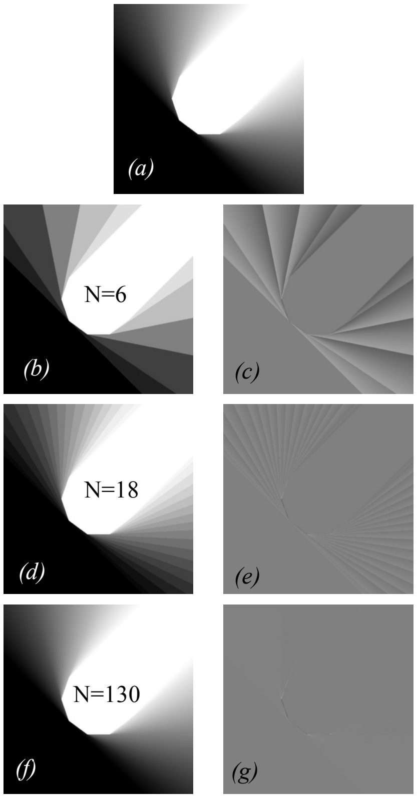

Figure 10. Error difference images. The reference object B is a decagon and the direction δ is π/4 (northeast). *(a)* The exact template values are computed for 65536 points of the Cartesian plane and stored in the form of an 8-bit, 256×256 gray level image. *(b)(d)(f)* Approximated templates are calculated and visualized through 8-bit, 256×256 gray level images. *(c)(e)(g)* Corresponding difference images. Medium gray, as in the bottom left parts of the images, indicates a zero error. Lighter or darker areas indicate that the template values have been overestimated or underestimated.

Whatever the direction δ and the reference object B, experiments confirm that the error SAD drops fast and then stabilizes as the number N of regions increases (Fig. 9). When a few tens of regions are considered, the template values are significantly over-estimated or underestimated (Fig. 10c). However, when more than fifty regions are considered, there is no noticeable difference between the approximated and exact values (Fig. 10g).

4.3 Computational Efficiency

This set of experiments was conducted on a 2.4GHz P4 CPU with 1024 MB physical memory, running Windows 2000. The implementation language was C++. As illustrated by Fig. 11, the processing time increases linearly with the number N of template regions, whether the reference object B is convex or concave. If B is convex, the processing time also increases linearly with m, the number of vertices of B. Theoretically, if B is concave, it might increase quadratically with m. Practically, however, unless fractal-like objects are considered, the increase is, once again, linear (or quasilinear). This is illustrated by Fig. 12. In both figures, the direction δ is π/4 (northeast).

5. CONCLUSIONS AND FUTURE WORK

A directional spatial relationship to a reference object can be represented by a spatial template. These templates (which are given different names in the literature) play an important role in object localization tasks. In previous work, we considered the case of 2D raster data. Here, we have examined the case of 2D vector data. We have designed, based on the concept of the F-template, an efficient and flexible approximation algorithm for spatial template calculation. This algorithm, which also draws from silhouette determination techniques developed for computer graphics applications, is of particular interest for spatial query processing in Geographic Information Systems. The templates are stored in a vector form. Expressions like "north of the museum" can be visualized, and regions where the relationship holds can be distinguished from regions where it does not hold. Different perceptions can be accommodated. All types of transitions, from the most abrupt to the smoothest ones, can be considered. Once the template associated with the expression is calculated, propositions like "the post office is located north of the museum" can be finely assessed. Moreover, various points of view, from most pessimistic to most optimistic, can be given. The objects at hand are not necessarily convex, and they are not approximated through their centroid or minimum bounding rectangle. We have assumed, however, that each object is defined by a single polygon, i.e., each object is connected, without holes, and with a clear boundary. In future work, we will show that our algorithm can easily be extended to handle fuzzy objects, objects with holes, and objects having several connected components. We will discuss the case of more complex expressions (e.g., "north of the museum and southeast of the park"). Eventually, we will also examine the case of 3D vector data.

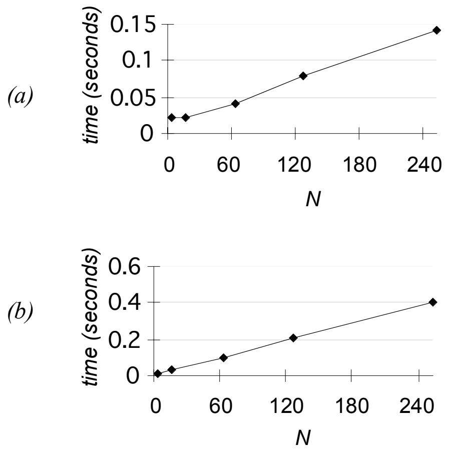

Figure 11. Relationship between processing time and number of regions. *(a)* The reference object is as in Fig. 9a. *(b)* The reference object is as in Fig. 9b.

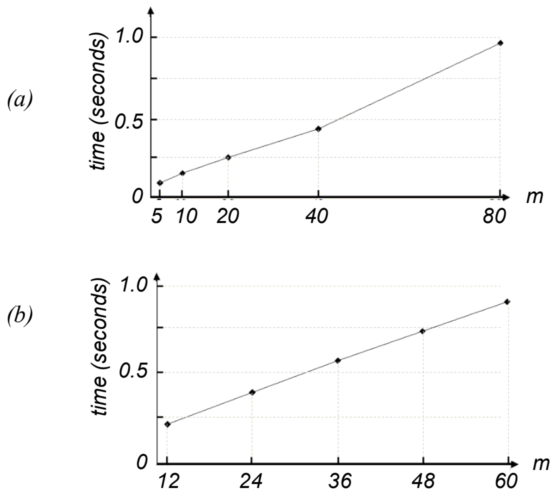

Figure 12. Relationship between processing time and number of vertices. *(a)* The reference object is a regular polygon (approximation of a disk). *(b)* The reference object is a cross-shaped object as in Fig. 9b, with additional vertices spread along its boundary.

6. ACKNOWLEDGMENTS

The authors want to express their gratitude for support from the Natural Science and Engineering Research Council of Canada (NSERC), grant 045638.

7. REFERENCES

[1] R. Krishnapuram, J.M. Keller, and Y. Ma, “Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions”, *IEEE Trans. on Fuzzy Systems*, 1993, vol. 1, no. 3, pp. 222-233

[2] J. Sharma, and D.M. Flewelling, “Inferences from Combined Knowledge about Topology and Direction”, *SSD'95 (Advances in Spatial Databases: 42nd Symposium)*, 1995, *Proceedings*, pp. 279-291

[3] D. Papadias, and Y. Theodoridis, “Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures”, *Int. J. of Geographical Information Science*, 1997, vol. 11, pp. 111-138

[4] P. Matsakis, and L. Wendling, “A New Way to Represent the Relative Position of Areal Objects”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1999, vol. 21, no. 7, pp. 634-643

[5] F. Petry, M. Cobb, D. Ali, R. Angryk, M. Paprzycki, S. Rahimi, L. Wen, and H. Yang, “Fuzzy Spatial Relationships and Mobile Agent Technology in Geospatial Information Systems”, in P. Matsakis, and L. Sztandera (Eds.), *Applying Soft Computing in Defining Spatial Relations*, Studies in Fuzziness and Soft Computing, Physica-Verlag, 2002, vol. 106, pp. 123-155

[6] D. Peuquet, and C.-X. Zhan, “An algorithm to determine the directional relationship between arbitrarily-shaped polygons in a plane”, *Pattern Recognition*, 1987, vol. 20, pp. 65-74

[7] R. K. Goyal, and M. J. Egenhofer, “Similarity of Cardinal Directions”, in C. Jensen, M. Schneider, B. Seeger, V. Tsotras (Eds.), *Seventh International Symposium on Spatial and Temporal Databases Lecture Notes in Computer Science*, 2001, vol. 2121, pp. 36-55

[8] C. Claramunt, and M. Theriault, “Fuzzy semantics for direction relations between composite regions”, *Information Science*, 2004, pp. 73-90

[9] A. U. Frank, “Qualitative spatial reasoning: Cardinal directions as an example”, *International Journal of Geographical Information Systems (10) 3*, 1996, pp. 269-290

[10] I. Bloch, “Fuzzy Relative Position Between Objects in Image Processing: A Morphological Approach”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1999, vol. 21, no. 7, pp. 657-664

[11] G.D. Logan, and D.D. Sadler, “A Computational Analysis of the Apprehension of Spatial Relations”, in P. Bloom, M.A. Peterson, L. Nadel, and M.F. Garrett (Eds.), *Language and Space*, Cambridge, MA: MIT Press, 1996

[12] K.-P. Gapp, “Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space”, *12th National Conf. on Artificial intelligence*, Seattle, Washington, 1994, *Proceedings*, vol. 2, pp. 1393-1398

[13] P. Olivier, and J.-I. Tsuji, “Quantitative Perceptual Representation of Prepositional Semantics”, in *Artificial Intelligence Review (Special Issue on Integration of Natural Language and Vision Processing)*, 1994, vol. 8, pp. 147-158

[14] P. Matsakis, J. Ni, X. Wang, “Object Localization Based on Directional Information: Case of 2D Raster Data”, *ICPR 2006 (Int. Conf. on Pattern Recognition)*, Hong-Kong, 2006

[15] J. Freeman, “The Modeling of Spatial Relations”, *Computer Graphics and Image Processing*, 1975, vol. 4, pp. 156-171

[16] K.-P. Gapp, “Angle, Distance, Shape, and Their Relationship to Projective Relations”, *17th Annual Conf. of the Cognitive Science Society*, Pittsburgh, PA, 1995, *Proceedings*, Mahwah, NJ: Lawrence Erlbaum, pp. 112-117

[17] N. Franklin, L.A. Henkel, and T. Zengas, “Parsing Surrounding Space Into Regions”, *Memory and Cognition*, 1995, vol. 23, pp. 397-407

[18] P. Matsakis, *Relations Spatiales Structurelles et Interprétation d’images*, PhD Thesis, Institut de Recherche en Informatique de Toulouse, France, 1998