Directional Force Field-based Maps: Implementation and Application

JingBo Ni, Melanie Veltman, Pascal Matsakis

Computing and Information Science, University of Guelph, ON, N1G 2W1, Canada
{jni@uoguelph.ca, mveltman@uoguelph.ca, pmatsaki@uoguelph.ca}

Abstract. A directional relationship (e.g., right, above) to a reference object can be modeled by a directional map – an image where the value of each point represents how well the relationship holds between the point and the object. As we showed in previous work, such a map can be derived from a force field created by the object (which is seen as a physical entity). This force field-based model, defined by equations in the continuous domain, shows unique characteristics. However, the approximation algorithms that were proposed in the case of 2-D raster data lack efficiency and accuracy. We introduce here new algorithms that correct this flaw, and we illustrate the potential of the force field-based approach through an application to scene matching.

Keywords: spatial relationships, force fields, directional maps, scene matching.

1 Introduction

Research on the modeling of spatial relationships raises two questions: (a) How to identify the relationships between two given objects [1,2]? (b) How to identify, in a scene, the object that best satisfies a given relationship to a reference object [3]? The second question defines an object localization task. One theory supported by cognitive experiments is that people accomplish this task by parsing space around the reference object into *good* regions (where the object being sought is more likely to be), *acceptable* and *unacceptable* regions (where the object being sought cannot be) [4,5]. These regions form a so-called *spatial template* [5,6], which assigns each point in space a value between 0 (unacceptable region) and 1 (good region).

When focusing on *directional* (also called *projective* [7] or *cardinal* [8]) relationships (e.g., front, south, above), spatial templates can be referred to as *directional maps* [9] (or as *fuzzy landscapes* [3]). A directional map is an image where the value of each point reflects the degree to which the point satisfies some directional relationship to a reference object.

The map as defined in [3] takes the object's shape into account and depends essentially on angular deviation (two characteristics supported by cognitive studies). We call it the *standard map*, $S^{\delta R}$, where δ represents the directional relationship and R the reference object. Two algorithms have been designed for fast calculation of $S^{\delta R}$:

the first one is based on a propagation technique [3] and the second one on partitioning the image into parallel lines [10].

Matsakis *et al.* [9] proposed another model (Section 2), which relies on the idea of considering the reference object as a physical entity that creates a force field. All directional maps induced by the object can then be derived from the force field. Compared with standard maps, force field-based maps better cope with outliers, elongated objects and concavities [9]. However, the algorithm for force field computation described in [9] is slow. Moreover, although the directional maps induced by the object can be derived from the force field in negligible time, their calculation lacks accuracy. Indeed, the maps depend on a supremum that can only be estimated, and which, in [9], is often greatly overestimated. We introduce here new algorithms that correct these flaws (Section 3) and we illustrate the potential of the force field-based approach through an application to scene matching (Section 4). Note that directional maps can be used for many tasks, including spatial reasoning, object localization and identification, structural and model-based pattern recognition [11,12,13,14].

2 Force Field-based Maps

The notations used in the rest of the paper are as follows. $\mathbb{Z}_+$ is the set of positive integers. $\mathbb{P}$ is the Euclidean plane. For any points p and q of $\mathbb{P}$, pq is the vector from p to q with norm $|pq|$. $\vec{\delta}$ is the unit vector pointing at direction $\delta \in [0,2\pi)$. The radian measure in $[0,\pi]$ of the angle between two nonzero vectors $\vec{u}$ and $\vec{v}$ is denoted by $\angle(\vec{u},\vec{v})$. An *object* R is a subset of $\mathbb{P}$, bounded, closed, with area $|R| \neq 0$. We have $R(p)=1$ if $p \in R$ and $R(p)=0$ if $p \notin R$.

We assume that point q exerts on point p a force of magnitude $1/|pq|^r$ in the direction of pq, where r is a given real number. The force $\Phi_r^R(q)$ that q exerts on R is the integration of the forces that q exerts on all the points of R:

$$\Phi_r^R(q) = \int_{p\in\mathbb{P}} \frac{R(p)}{|pq|^r} \frac{pq}{|pq|} \mathrm{d}p = \int_{p\in R} \frac{pq}{|pq|^{r+1}} \mathrm{d}p \,. \qquad (\mathbf{1})$$

Φ_r^R is called the *force field* created by R [9]. Note that the algorithm for force field computation described in [9] is rather slow. A much more efficient algorithm is introduced in Section 3.1. The *force field-based map* $\Phi_r^{\delta R}$ in direction δ can be defined by, e.g., $\Phi_r^{\delta R}(q)=\mu(\angle(\Phi_r^R(q),\vec{\delta}))$, with $\mu(x)=\max\{0,1-2x/\pi\}$, or:

$$\Phi_r^{\delta R}(q) = \max\{0, (\Phi_r^R(q)\cdot\vec{\delta}) / (\sup_{p\in\mathbb{P}} \Phi_r^R(p)\cdot\vec{\delta})\} \,. \qquad (\mathbf{2})$$

We focus here on (2), which explicitly takes account of distance information (when $r \neq 0$). Unfortunately, $\sup_{p\in\mathbb{P}} \Phi_r^R(p)\cdot\vec{\delta}$ cannot be easily determined (unless $r=0$). In [9], Matsakis *et al.* replaced it with an upper bound that was determined analytically. However, this upper bound is often much higher than the supremum itself. As a result, the calculated value for $\Phi_r^{\delta R}(q)$ is often unreasonably low. In Section 3.2, we show that $\sup_{p\in\mathbb{P}} \Phi_r^R(p)\cdot\vec{\delta}$ can be better estimated using a heuristic search algorithm.

3 Implementation

Here, we see an object R in a digital image $\mathbb{G}$ as a surface covered by pixels (of size 1×1), not as a discrete cloud of points like in [9]. The force that one pixel exerts on another is defined based upon the following considerations. First, we draw a horizontal axis X and a vertical axis Y. For any two distinct points p and q of $\mathbb{P}$, let $\theta\in[0,2\pi)$ be the angle between pq and X, p_X and q_X be the projections of p and q on X, and p_Y and q_Y be their projections on Y. Thus, the magnitude of the force that q exerts on p can also be computed as: $1/|pq|^r = |\cos\theta|^r/|p_Xq_X|^r$ if $\theta\in\Theta_h=[0,\pi/4]\cup[3\pi/4,5\pi/4]\cup[7\pi/4,2\pi)$; and $1/|pq|^r=|\sin\theta|^r/|p_Yq_Y|^r$ if $\theta\in\Theta_v=(\pi/4,3\pi/4)\cup(5\pi/4,7\pi/4)$. See Fig. 1(a). Now, consider p and q two distinct pixels in $\mathbb{G}$ centered at points (x_p,y_p) and (x_q,y_q). We define that the force pixel q exerts on pixel p is in the direction of (x_q-x_p,y_q-y_p), and has the magnitude: $F_r(p,q)=|\cos\theta|^r f_r(|x_q-x_p|)$ if $\theta\in\Theta_h$; and $F_r(p,q)=|\sin\theta|^r f_r(|y_q-y_p|)$ if $\theta\in\Theta_v$, where θ is the angle between (x_q-x_p,y_q-y_p) and X. Now, let us only consider the case that $\theta\in\Theta_h$. The value of $1/|p_Xq_X|^r$ mentioned above in fact is the magnitude of the force between the projections of the points p and q on X, p_X and q_X, which are points too. When p and q are pixels, their projections on X, I and J, are unit line segments (instead of points), as shown in Fig. 1(b). It is therefore natural to define $f_r(s)$, $s\in\mathbb{Z}_+$, as the sum of the forces that the points of J exert on the points of I, i.e.,

$$f_r(s)=\int_0^1\int_0^1 1/(y-x+s)^r\,\mathrm{d}x\mathrm{d}y\,. \tag{3}$$

Function f_r is well defined on $\mathbb{Z}_+$ when $r<2$, and the double integral can be solved analytically. When $p=q$, we set $F_r(p,q)=0$ indicating that the forces that a pixel exerts on itself are balanced out. Having defined the force between two pixels, the force that one pixel q exerts on a raster object R (i.e., a set of pixels) can be computed as:

$$\Phi_r^R(q)=\sum_{p\in\mathbb{G}} R(p)F_r(p,q)(x_q-x_p,y_q-y_p)/|(x_q-x_p,y_q-y_p)|\,. \tag{4}$$

3.1 An Algorithm for Approximating Φ_r^R

For any image $\mathbb{G}$ of size $N=m\times n$, Equation (4) calculates $\Phi_r^R(q)$ in $\mathcal{O}(N)$, i.e., it calculates the entire force field Φ_r^R in $\mathcal{O}(N^2)$. Here, we propose an algorithm for fast approximating Φ_r^R. Let q be the origin of $\mathbb{P}$, i.e., $q=(0,0)$. By rewriting (1) using the polar coordinates (θ,ℓ) of p, we have:

$$\Phi_r^R(q)=\Phi_r^R(0,0)=\int_0^{2\pi}\int_0^{+\infty} R(\theta,\ell)(-\cos\theta,-\sin\theta)/\ell^{r-1}\mathrm{d}\ell\mathrm{d}\theta\,. \tag{5}$$

By letting $\ell=s/|\cos\theta|$ for $\theta\in\Theta_h$ and $\ell=s/|\sin\theta|$ for $\theta\in\Theta_v$, (5) becomes:

$$\begin{aligned}\Phi_r^R(q)=&\int_{\theta\in\Theta_h}|\cos\theta|^{r-2}(-\cos\theta,-\sin\theta)[\int_0^{+\infty}R(\theta,s/|\cos\theta|)/s^{r-1}\mathrm{d}s]\mathrm{d}\theta\\&+\int_{\theta\in\Theta_v}|\sin\theta|^{r-2}(-\cos\theta,-\sin\theta)[\int_0^{+\infty}R(\theta,s/|\sin\theta|)/s^{r-1}\mathrm{d}s]\mathrm{d}\theta\end{aligned}\,. \tag{6}$$

In (6), point $p=(\theta,s/|\cos\theta|)$ (or $(\theta,s/|\sin\theta|)$) lies on the line that starts from q and points at direction θ, and $1/s^{r-1}$ is the magnitude of the force between the projects of p and q on axis X (or Y). According to the discussion made above, we can transform (6) to

$$\begin{aligned}\Phi_r^R(q) = &\sum_{\theta\in\Theta_h} |\cos\theta|^{r-2}\,(-\cos\theta,-\sin\theta)[\sum_{s\in\mathbb{Z}_+} R(p_{\theta,s}) f_{r-1}(s)]\Delta\theta \\ &+\sum_{\theta\in\Theta_v} |\sin\theta|^{r-2}\,(-\cos\theta,-\sin\theta)[\sum_{s\in\mathbb{Z}_+} R(p_{\theta,s}) f_{r-1}(s)]\Delta\theta\end{aligned}, \tag{7}$$

for handling raster data. In (7), θ belongs to a set $\{2\pi k/K\}_{k\in 0..K-1}$ of K reference directions, and therefore $\Delta\theta$ is $2\pi/K$. $p_{\theta,0}=q$, $p_{\theta,1}$, $p_{\theta,2}$, etc., are the pixels successively encountered on the rasterization of the line $\Lambda_\theta(\omega)$, which starts from an edge pixel ω of $\mathbb{G}$ and points at direction θ. See Fig. 1(c). Since R is crisp, which means each $R(p_{\theta,s})$ is either 1 or 0, expression $\sum_{s\in\mathbb{Z}_+} R(p_{\theta,s}) f_{r-1}(s)$ in (7) can then be written as:

$$\sum_{s\in\mathbb{Z}_+} R(p_{\theta,s}) f_{r-1}(s) = \sum_{i=1}^{M}\sum_{s=a_i}^{b_i} f_{r-1}(s) = \sum_{i=1}^{M}\int_{a_i}^{b_i+1}\int_0^1 1/(y-x)^{r-1}\,\mathrm{d}x\mathrm{d}y\,. \tag{8}$$

M here is the number of segments of R on the line $\Lambda_\theta(\omega)$ encountered after q. When R is convex, $M\le 1$ for any θ and q. In (8), the rightmost double integral can again be analytically solved. Equations (7,8) then calculate $\Phi_r^R(q)$ in $\mathcal{O}(KM)$, and calculate Φ_r^R in $\mathcal{O}(KMN)+\mathcal{O}(KN)=\mathcal{O}(KMN)$, where $\mathcal{O}(KN)$ time is required to rasterize the lines $\Lambda_\theta(\omega)$ (for all θ and ω) in $\mathbb{G}$, and to determine the segments of R on those lines. In practice, the value of M is usually small, however, in the worst case, the value of M can reach $\sqrt{N}$, which raises the complexity up to $\mathcal{O}(KN\sqrt{N})$. When R is fuzzy, the manipulation of R can always be reduced to that of its level-cuts, which are crisp.

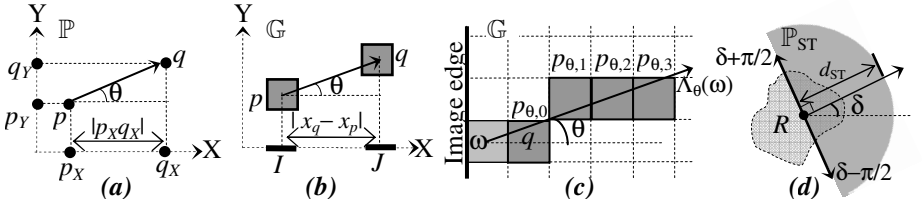


Fig. 1. *(a)* The force between two points; *(b)* The force between two pixels; *(c)* The rasterization of the line $\Lambda_\theta(\omega)$; *(d)* The searching territory $\mathbb{P}_{ST}$.

3.2 Estimation of $\sup_{p\in\mathbb{P}} \Phi_r^R(p)\cdot\vec{\delta}$

When $r=0$, we have $\sup_{p\in\mathbb{P}} \Phi_0^R(p)\cdot\vec{\delta} = |R|$ [9]. For $r\neq 0$, we develop an algorithm which searches in a pre-determined territory for the point $p_{\max}$, such that $\Phi_r^R(p_{\max})\cdot\vec{\delta}$ forms a good approximation of $\sup_{p\in\mathbb{P}} \Phi_r^R(p)\cdot\vec{\delta}$. Equation (2) can then be replaced with:

$$\Phi_r^{\delta R}(q) = \max\{0, \min\{1, (\Phi_r^R(q)\cdot\vec{\delta}) / (\Phi_r^R(p_{\max})\cdot\vec{\delta})\}\}\,. \tag{9}$$

For easy illustration, we express $\mathbb{P}$ in terms of polar coordinates (θ,ℓ) and set the origin (0,0) at the centroid of R. The searching territory (Fig. 1(d)) is defined as: $\mathbb{P}_{ST} = [\delta-\pi/2,\delta+\pi/2]\times[0,d_{ST}]\subset\mathbb{P}$. $d_{ST}=\alpha+\beta/\sqrt{r}$, where $\alpha=2\int_\theta\int_\ell R(\theta,\ell)|\ell\cos(\theta-\delta)|/|R|\mathrm{d}\ell\mathrm{d}\theta$ and $\beta=2\int_\theta\int_\ell R(\theta,\ell)|\ell\sin(\theta-\delta)|/|R|\mathrm{d}\ell\mathrm{d}\theta$, is the pre-determined searching distance. The searching algorithm is given as follows:

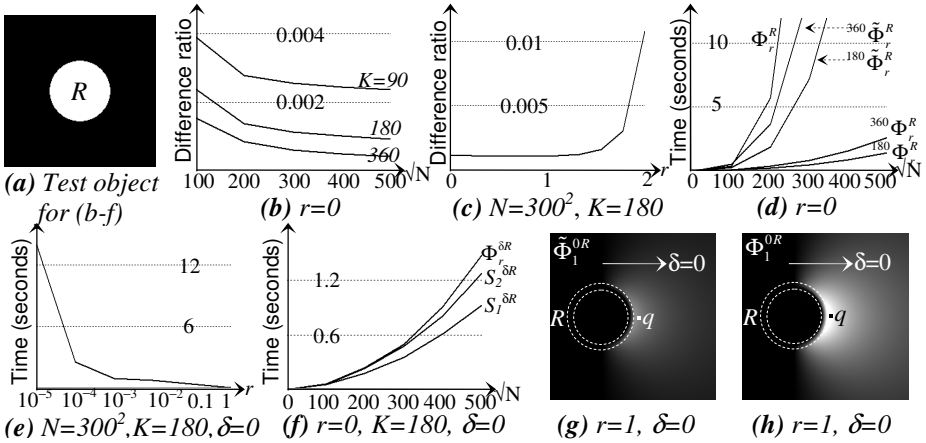
δ′=δ; $d=d_{ST}$; /*The initial searching direction and distance are δ and d_{ST}.*/
s_p=5; /*The searching speed is set to 5*/
Δδ=0.001; /* the minimum angle difference is set to 0.001*/
FOR each iteration /*The number of iterations is set to 5*/
δ_a=δ′−π/2, δ_b=δ′+π/2;
WHILE $\delta_b-\delta_a \geq \Delta\delta$ /*Search on the curve defined by $[\delta_a,\delta_b]\times\{d\}$ */
$p_a=(\delta_a,d)$ and $p_b=(\delta_b,d)$;
IF $\Phi_r^R(p_a)\cdot\vec{\delta} < \Phi_r^R(p_b)\cdot\vec{\delta}$: $\delta_a=\delta_a+(\delta_b-\delta_a)/s\_p$; **ELSE**: $\delta_b=\delta_b-(\delta_b-\delta_a)/s\_p$;
δ′=δ_b, d_a=0 and $d_b=d_{ST}$; /*δ′ is the adjusted searching direction*/
WHILE $d_b - d_a \geq 1$ /*Search on the line defined by $\{\delta'\}\times[d_a,d_b]$ */
$p_a=(\delta',d_a)$ and $p_b=(\delta',d_b)$;
IF $\Phi_r^R(p_a)\cdot\vec{\delta} < \Phi_r^R(p_b)\cdot\vec{\delta}$: $d_a=d_a+(d_b-d_a)/s\_p$; **ELSE**: $d_b=d_b-(d_b-d_a)/s\_p$;
$d=d_b$; /*d is the adjusted searching distance*/
IF $sup < \Phi_r^R(p_b)\cdot\vec{\delta}$: $sup = \Phi_r^R(p_b)\cdot\vec{\delta}$; /* sup is initially set to 0 */
ELSE: RETURN $p_{max}=p_b$; /* Nothing to update means we found $p_{max}=p_b$ */
RETURN $p_{max}=p_b$; /*After all iterations, we let $p_{max}=p_b$ */

3.3 Experiments

Let Φ_r^R be the exact force field calculated according to (4), and ${}^K\Phi_r^R$ be the force field computed using (7,8). The difference between Φ_r^R and ${}^K\Phi_r^R$ is measured by the difference ratio (*DR*), which takes on values in [0,1], and is 0 *iff* $\Phi_r^R = {}^K\Phi_r^R$:

$$DR = (\sum_q |\Phi_r^R(q) - {}^K\Phi_r^R(q)|) / \sum_q (|\Phi_r^R(q)| + |{}^K\Phi_r^R(q)|) . \quad (\mathbf{10})$$

The force field ${}^K\Phi_r^R$ approximates Φ_r^R . The accuracy of the approximation increases with *K* (Fig. 2(b)) and is quite high (*DR* is less than 0.5%) even when *K* is relatively small (*K*=90). The accuracy also depends on the image size *N* and on *r* (Figs. 2(b,c)).



***(a)** Test object for (b-f)* ***(b)** r=0* ***(c)** $N=300^2$, K=180* ***(d)** r=0*
***(e)** $N=300^2$,K=180, δ=0* ***(f)** r=0, K=180, δ=0* ***(g)** r=1, δ=0* ***(h)** r=1, δ=0*

Fig. 2. Experiments. $S_1^{\delta R}$ and $S_2^{\delta R}$ in *(f)* are the standard maps generated using the first [3] and the second algorithms [10] (Section 1). All algorithms were implemented in C.

Let ${}^{K}\tilde{\Phi}_r^R$ be the force field computed using the algorithm proposed in [9]. Fig. 2(d) shows that the processing times of ${}^{K}\tilde{\Phi}_r^R$ and ${}^{K}\Phi_r^R$ both increase with K and N, but at different rates. Computing ${}^{K}\Phi_r^R$ is far more efficient than computing ${}^{K}\tilde{\Phi}_r^R$ and Φ_r^R. Once the force field is calculated, directional maps $\Phi_r^{\delta R}$ in various δ can be computed using (9). The Computation is generally fast, unless r takes some arbitrarily small positive value (like 10^{-5}) (Fig. 2(e)). Fig. 2(f) shows that the computation of $\Phi_r^{\delta R}$ (${}^{K}\Phi_r^R$ plus $\Phi_r^{\delta R}$) is comparably efficient to that of the standard map $S^{\delta R}$.

Let $\tilde{\Phi}_r^{\delta R}$ be the force field-based map generated by the second transformation proposed in [9]. Consider the case that R is a concentric shell, r=1, and δ=0. The values of $\tilde{\Phi}_1^{0R}$ are all fairly low ($\tilde{\Phi}_1^{0R}$ appears dark in Fig.2 (g)) due to the overestimation problem mentioned in Section 2. This raises some serious issues. For example, according to intuition, pixel q (in Fig.2 (g)) is perfectly close to and to the right of R. However, $\tilde{\Phi}_1^{0R}$ somewhat denies this perception since $\tilde{\Phi}_1^{0R}(q)$ =0.5. In the map generated using (9), Φ_1^{0R} (Fig.2 (h)), such issue does not exist and $\Phi_1^{0R}(q)$ =1.

4 Application

Here, we illustrate the potential of the force field-based approach through a scene matching task. Consider a target scene depicting a number of objects. As an example, the scene in Fig. 3(a) contains 21 disconnected objects, Figs. 3(b,c) show two (hand-drawn) query scenes, and the task is to determine if there exists a match between query and target. Note that for our purposes, a 'match' exists when there are objects in the target scene whose relative positions correspond to those found between the objects in the query scene. Furthermore, we want matching to be invariant to scaling and rotation. What follows is a description of how this task can be performed.

Consider a reference object R and a number of located objects L_i with i=1..n. Object R's view histogram in direction δ, h_R^δ, is a function from $\{t_k=k/P\}_{k=0..P}$ to [0,1]:

$$h_R^\delta(t_k) = \sum_{\{q \mid |\Phi_r^{\delta R}(q)-t_k|=\min_j |\Phi_r^{\delta R}(q)-t_j|\}} [(\sum_{i=1}^{n} L_i(q)) / \sum_{i=1}^{n} |L_i|] . \quad (11)$$

$h_R^\delta(t_k)$ counts (in a normalized way) the pixels q in the located objects such that $\Phi_r^{\delta R}(q)$ is best approximated by t_k. Now, assume there are n objects in the query scene Q. One of them is selected as the reference object R, the others are the located objects. Here, R is the object whose centroid is closest to the centroid of the entire scene. Then, for each view direction $\delta_i=2\pi i/D$ with i=0..D−1, four view histograms of R are computed in Q:

$$Q_R^{\delta_i} = (h_R^{\delta_i}, h_R^{\delta_i+\pi/2}, h_R^{\delta_i+\pi}, h_R^{\delta_i+3\pi/2}) . \quad (12)$$

Assume there are $m \geq n$ objects in the target scene T. The matching between T and Q is conducted in an exhaustive way:

FOR each object O in T:

  Let O be the reference object;

  List all the possible ways of drawing n−1 objects from the other m−1 objects in T;

FOR each drawing:
Let the $n-1$ objects be the located objects, which, together with the reference object O, form a sub-scene T' of T;
Compute O's view histograms in the sub-scene: $T'_O = (h_O^0, h_O^{\pi/2}, h_O^{\pi}, h_O^{3\pi/2})$;
FOR each $Q_R^{\delta_i}$:
Compute the similarity degree between T'_O and $Q_R^{\delta_i}$, $sim(T'_O, Q_R^{\delta_i})$.
Let $\delta_{max}=\delta_k$ such that $sim(T'_O, Q_R^{\delta_k})$ = $\max_i\{ sim(T'_O, Q_R^{\delta_i}) \}$, and **Let** $sim(T'_O, Q_R^{\delta_{max}})$ be the degree of matching between Q and T'.

The similarity degree between T'_O and $Q_R^{\delta_i}$, $sim(T'_O, Q_R^{\delta_i})$, is computed as:

$$sim(T'_O, Q_R^{\delta_i}) = \min_{j=0..3}\{d(h_O^{j\pi/2}, h_R^{\delta_i + j\pi/2})\}, \quad (13)$$

where $d(h_1,h_2)=\max\{0,1-\sum_k|h_1(t_k)-h_2(t_k)|\}$ measures the similarity between the view histograms h_1 and h_2. Note that $d(h_1,h_2)=1$ *iff* $h_1=h_2$. Finally, we present the sub-scenes T' that best match Q (the sub-scenes with highest degrees $sim(T'_O, Q_R^{\delta_{max}})$).

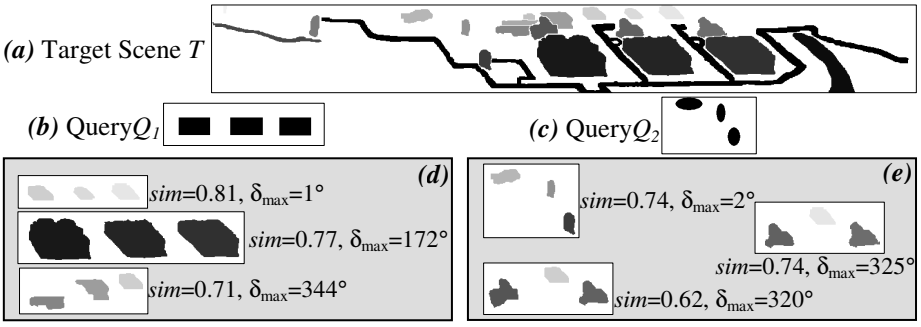


Fig. 3. A scene matching task. *(a)* A hand-segmented laser radar range image of the power-plant at China Lake, CA. The image was used by Matsakis, Keller *et al.* in [15]. *(d)* The sub-scenes of T that best match Q_1, and *(e)* those that best match Q_2, where *sim* is the similarity degree between a query and a sub-scene (after counterclockwise rotation by δ_{max}).

As shown by Fig. 3, the proposed algorithm generates reasonable results. In this experiment, P=100, D=360, and each force field-based map was computed using the algorithm defined by (7,8,9) with r=0 and K=90. Under this configuration, for each of the query scenes in Figs. 3(b,c), the matching algorithm finishes within one minute. Of course, smaller values of P, D and K can be chosen to compromise precision for speed. Readers may ask whether standard maps can be applied to scene matching tasks like the one presented here. The answer is negative. Since standard maps rely merely on angular deviation, they lack representation power for dimension and distance information, which is critical to these tasks.

5 Conclusions

In [9], Matsakis *et al.* developed a new quantitative model of the directional relationships to a reference object. The model relies on the idea that all directional

maps induced by the object can be derived from a force field. However, the proposed algorithms lack efficiency and accuracy. In this paper, we have introduced algorithms that correct this flaw and we have demonstrated the potential of the force field-based approach through an application to scene matching. In future work, we will further explore the idea of using directional maps as a tool for pattern recognition and scene understanding.

Acknowledgements

The authors want to express their gratitude for support from the Natural Science and Engineering Research Council of Canada (NSERC), grant 262117.

References

1. Krishnapuram, R., Keller, J.M., Ma, Y.: Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions. IEEE Trans. on Fuzzy Systems, 1(3), 222-233 (1993)
2. Matsakis, P., Wendling, L.: A New Way to Represent the Relative Position between Areal Objects. IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(7), 634-643 (1999)
3. Bloch, I.: Fuzzy Relative Position Between Objects in Image Processing: A Morphological Approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(7), 657-664 (1999)
4. Franklin, N., Henkel, L.A., Zangas, T.: Parsing Surrounding Space into Regions. Memory & Cognition, 23(4), 397-407 (1995)
5. Logan, G.D., Sadler, D.D.: A Computational Analysis of the Apprehension of Spatial Relations. Language and Space, Cambridge, MA: MIT Press, 493-529 (1996)
6. Carlson-Radvansky, L.A., Logan, G.D.: The Influence of Reference Frame Selection on Spatial Template Construction. Memory & Language, 37(3), 411-437 (1997)
7. Gapp, K.-P.: Angle, Distance, Shape, and Their Relationship to Projective Relations. Proc. 17th Conf. Cognitive Science Soc., 112-117 (1995)
8. Frank, A.U.: Qualitative Spatial Reasoning: Cardinal Directions as an Example. Int. J. of Geographical Information Systems, 10(3): 269-290 (1996)
9. Matsakis, P., Ni, J., Veltman, M.: Directional Relationships to a Reference Object: A Quantitative Approach based on Force Fields. Submitted to ICIP 09.
10. Matsakis, P., Ni, J., Wang, X.: Object Localization based on Directional Information: Case of 2D Raster Data. Proc. 18th Int. Conf. on Pattern Recognition, 2, 142-146 (2006)
11. Bloch, I., Saffiotti, A.: On the Representation of Fuzzy Spatial Relations in Robot Maps. In: Bouchon-Meunier, B., Foulloy, L., Yager, R.R. (eds.) Intelligent Systems for Information Processing, 47-57, Elsevier, NL (2003)
12. Colliot, O., Camara, O., Bloch, I.: Integration of Fuzzy Spatial Relations in Deformable Models-Application to Brain MRI Segmentation. Pattern Recognition, 39, 1401-1414 (2006)
13. Krishnapuram, R., Medasani, S., Jung, S.-H., Choi, Y.-S, Balasubramaniam, R.: Content-based Image Retrieval Based on a Fuzzy Approach. IEEE Trans. on Knowledge and Data Engineering, 16(10): 1185-1199 (2004)
14. Smith, G.B., Bridges, S.M.: Fuzzy Spatial Data Mining. Proc. NAFIPS, 184-189 (2002)
15. Matsakis, P., Keller, J., Wendling, L., Marjamaa, J., Sjahputera, O.: Linguistic Description of Relative Positions in Images. TSMC Part B, 31(4): 573-588 (2001)