

# QUANTITATIVE REPRESENTATION OF THE RELATIVE POSITION BETWEEN 3D OBJECTS

JingBo Ni, Pascal Matsakis, Lukasz Wawrzyniak  
Department of Computing and Information Science  
University of Guelph, ON, N1G 2W1  
Canada

jni@uoguelph.ca, matsakis@cis.uoguelph.ca, lwawrzyn@uoguelph.ca

## ABSTRACT

Spatial relationships play an important role in many domains of computer science, including computer vision, Geographic Information Systems (GIS), and medical imaging. In previous work, we introduced the notion of the histogram of forces. It is a quantitative representation of the relative position between two objects. It is sensitive to the shape, size, and orientation of the objects. The objects considered so far could be disjoint or intersecting, they could be bounded or unbounded, convex or concave, available in raster or in vector form, but they had to be two-dimensional. In this paper, we show that three-dimensional raster data can be handled as well. By adopting proper optimization procedures, the presented technique provides a fast and reliable way for representing the relative position between two 3D objects. The results of experiments conducted on synthetic and real data validate our approach.

## KEY WORDS

Relative position, spatial relations, force histograms, angle histograms, computer vision.

## 1. Introduction

The spatial organization of a scene is often expressed in terms of the spatial relationships between the objects or regions it contains. Representations that capture the essential attributes of spatial relationships between objects are therefore of critical importance in many fields, such as pattern recognition, spatial reasoning, scene understanding, spatial databases, and more.

Relative positions are often assessed by considering only some very simple geometric characteristics. For instance, the object can be approximated by its centroid [1], its minimum bounding rectangle [2], its minimum bounding box [3] or bounding right parallelepiped [4]. All of these methods suffer from oversimplification, as they do not take into account the true shape of the object.

The notion of the histogram of angles was introduced by Krishnapuram et al. [5] and Miyajima and Ralescu [6]. It is a shape-sensitive representation of the relative position between two objects. The calculation of the

histogram of angles relies on the calculation of angles defined by all pixel pairs (for 2D objects) or voxel pairs (for 3D objects). This computation, however, is expensive, particularly in the case of three-dimensional data. Furthermore, histograms of angles cannot be computed from vector data.

In [7] and [8], Matsakis and Wendling proposed the notion of the histogram of forces. In the 2D case, the force histogram generalizes and supersedes the angle histogram. It encapsulates structural information about the objects as well as information about their spatial relationships. It is sensitive to the shape of the objects, their orientation and their size. It is also sensitive to the distance between them and allows explicit and variable accounting of metric information. Moreover, the force histogram offers solid theoretical guarantees and nice geometric properties. It ensures fast and efficient processing of vector data as well as of raster data, and enables the handling of fuzzy objects as well as of crisp objects, intersecting objects as well as of disjoint objects, and unbounded objects as well as of bounded objects. The applications are numerous. For instance, relative positions can be described in terms of spatial relationships modeled by fuzzy relations. They can also be represented by linguistic expressions, for scene description or human-robot communication in natural language [9].

In this paper, the notion of the histogram of forces is extended to handle three-dimensional objects. Like in the 2D case, a set of evenly distributed reference directions is considered. It is generated using a random-start hill-climbing heuristic. Then, to lighten the computational burden, irrelevant directions are detected and dropped before further processing. Forces in the remaining directions are computed very much like in the 2D case.

The remainder of the paper is organized as follows. In Section 2, we briefly explain how angle histograms and force histograms can be associated with two-dimensional objects. In Section 3, we show that 3D objects can be handled as well. Section 4 presents examples and a comparative study where force histograms are matched against angle histograms. Conclusions are given in Section 5.

## 2. The Case of 2D Objects

In this section, the symbols  $A$  and  $B$  denote two 2D objects in the plane,  $n$  is a positive integer multiple of 4, and  $D^n$  is the set  $\{2\pi i/n\}_{i \in 0..n-1}$ . The values  $2\pi i/n$  define directions, called *reference directions*. They are evenly distributed in the plane and include the four primitive directions  $0$  (horizontal right),  $\pi/2$  (vertical above),  $\pi$  (horizontal left) and  $3\pi/2$  (vertical below). Moreover, if  $\theta$  belongs to  $D^n$ , then the opposite direction  $\theta+\pi$  (or  $\theta-\pi$ ) also belongs to  $D^n$ .

### 2.1 Angle Histograms

Here, we assume that  $A$  and  $B$  are available in raster form, i.e.,  $A$  and  $B$  are two sets of pixels. The histogram of angles associated with the pair  $(A,B)$  represents the position of  $A$  relative to  $B$ . It is a function  $\mathcal{A}^{AB}$  from  $\mathfrak{R}$  (the set of real numbers) into  $\mathfrak{R}_+$  (the set of non-negative real numbers). For any two pixels  $p$  and  $q$  with  $p \neq q$ , let  $qp$  be the oriented line that runs from the center of  $q$  to the center of  $p$ . For each  $\theta \in \mathfrak{R}$ , the value  $\mathcal{A}^{AB}(\theta)$  is the number of pixel pairs  $(p,q) \in A \times B$  such that the direction of  $qp$  is  $\theta$ . In practice, however, an *a posteriori* digitization of the set of directions needs to be carried out.  $\mathcal{A}^{AB}(\theta)$  is defined for reference directions only (i.e., for values  $\theta \in D^n$ ): it is the number of pixel pairs  $(p,q) \in A \times B$  such that the direction of  $qp$  is best approximated by  $\theta$  (among all reference directions). Note that all pixel pairs have to be considered, whatever the number  $n$  of reference directions. Therefore,  $n$  has a negligible impact on the processing time of the angle histogram. One example is shown in Fig. 1.

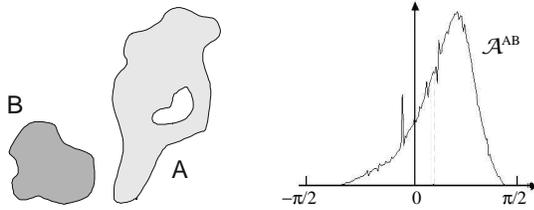


Fig. 1. Angle histograms. In this example, 360 reference directions were considered ( $n=360$ ).

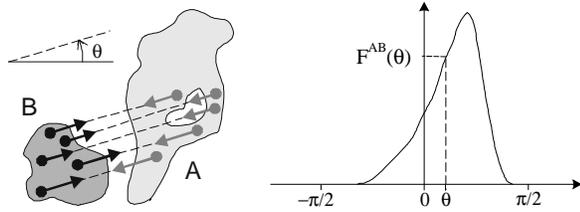


Fig. 2. Force histograms.  $F^{AB}(\theta)$  is the scalar resultant of forces that tend to move  $B$  in direction  $\theta$  (black arrows). In this example,  $F^{AB}$  is the histogram of constant forces. 360 reference directions were considered ( $n=360$ ).

### 2.2 Force Histograms

A histogram of forces  $F^{AB}$  is also a function from  $\mathfrak{R}$  into  $\mathfrak{R}_+$  that represents the position of an object  $A$  relative to another object  $B$ . For any direction  $\theta$ , the value  $F^{AB}(\theta)$  is the scalar resultant of elementary forces. These forces are exerted by the points of  $A$  on those of  $B$ , and each tends to move  $B$  in direction  $\theta$  (Fig. 2). You can imagine, for instance, gravitational forces between two flat metal plates of uniform density and constant and negligible thickness. Here, however, we are not limited by the laws of physics. If the elementary forces do not depend on the distance between the points considered (i.e., constant forces) then  $F$  is denoted by  $F_0$ . It can be shown [7][8] that the histogram of constant forces coincides with the histogram of angles, but without its weaknesses.

In practice, an *a priori* digitization of the set of directions is carried out.  $F^{AB}(\theta)$  is computed for the  $n$  reference directions only, and the processing time is proportional to  $n$ . Actually, the letter  $F$  denotes a numerical function. For any oriented line  $\Delta$ , the value  $F(A \cap \Delta, B \cap \Delta)$  is the scalar resultant of elementary forces. These forces are exerted by the points of  $A \cap \Delta$  on those of  $B \cap \Delta$ , and each tends to move  $B$  in the direction pointed by  $\Delta$ . You can imagine forces between aligned straight metal rods of uniform density and constant and negligible diameter. In the case of raster data, each rod represents a set of adjacent pixels (of either  $A$  or  $B$ ) that are batch processed, and the computation of each  $F^{AB}(\theta)$  comes down to the computation of a finite number of values  $F(A \cap \Delta, B \cap \Delta)$ . Note that the function  $F$  is defined by a set of algebraic expressions that are determined using integral calculus, and then hard coded [7][8].

## 3. The Case of 3D Objects

In this section, the symbols  $A$  and  $B$  denote two 3D objects available in raster form, i.e.,  $A$  and  $B$  are two sets of voxels. From both theoretical and practical points of view, the notion of the histogram of angles can easily be extended to handle such objects. In this section, we will limit our discussion to the extension of the notion of the histogram of forces. The principle is the same. For any direction  $\vec{\theta}$  (now defined by a unit vector), the value  $F^{AB}(\vec{\theta})$  is the scalar resultant of elementary forces. These forces are exerted by the points of  $A$  on those of  $B$ , and each tends to move  $B$  in direction  $\vec{\theta}$ . Only a finite number of values  $F^{AB}(\vec{\theta})$  are considered:  $\vec{\theta}$  is taken from some set  $D^n$  of reference directions. The computation of each  $F^{AB}(\vec{\theta})$  comes down to the computation of a finite number of values  $F(A \cap \Delta, B \cap \Delta)$ , where  $\Delta$  denotes an oriented line and  $F$  the same numerical function as in Section 2.2. From a practical standpoint, however, the problem is not as simple: 3D raster data tends to be much more voluminous than 2D data, and proper optimization methods have to be adopted.

### 3.1 Reference Directions

The  $n$  reference directions should satisfy the following constraints:

- They should be evenly distributed in the space.
- They should include the six primitive directions  $(\pm 1, 0, 0)$  (right/left),  $(0, \pm 1, 0)$  (above/below) and  $(0, 0, \pm 1)$  (front/behind).
- If  $\vec{\theta}$  belongs to the set  $D^n$  of reference directions, then the opposite direction  $-\vec{\theta}$  should also belong to  $D^n$ .

These constraints imply that  $n=6+8m$ , where “6” corresponds to the 6 primitive directions, “8” to the 8 regions delimited by the XY, XZ and YZ planes, and “m” to the number of reference directions in each region. Contrary to the 2D case, the term “evenly distributed” can be given different meanings and, for most values  $n$ , the distribution of  $n$  directions in the 3D space cannot be perfectly even. Bourke describes a simple random-start hill-climbing heuristic for spreading points evenly on a sphere [10]. First, points are randomly generated on the sphere. The points repulse each other, and an iterative process allows a stable configuration to be found. We slightly modified this heuristic to populate  $D^n$  and make sure that the second and third constraints are satisfied. Examples are shown in Fig. 3.

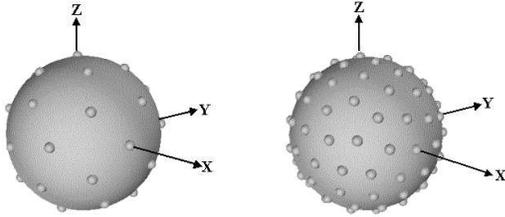


Fig. 3. Two sets of reference directions:  $D^{38}$  (left) and  $D^{110}$  (right).

### 3.2 Relevant Directions

A reference direction  $\vec{\theta}$  is *relevant* if  $F^{AB}(\vec{\theta}) \neq 0$ , i.e., if there exists an oriented line  $\Delta$  whose direction is  $\vec{\theta}$  and such that  $A \cap \Delta \neq \emptyset$  and  $B \cap \Delta \neq \emptyset$ . Otherwise, the direction is *irrelevant*, and need not be considered when computing  $F^{AB}$ . A simple and rapid test allows us to detect most of the irrelevant directions and, therefore, save computational time. Consider a reference direction  $\vec{\theta} = (x, y, z)$  and the plane  $\vec{\theta}^{\max}$  defined as follows:

- If  $\max\{x, y, z\} = x$  then  $\vec{\theta}^{\max}$  is the YZ plane.
- If  $\max\{x, y, z\} = y$  then  $\vec{\theta}^{\max}$  is the XZ plane.
- If  $\max\{x, y, z\} = z$  then  $\vec{\theta}^{\max}$  is the XY plane.

Now, consider the Minimum Bounding Boxes (MBBs) of the two objects  $A$  and  $B$  and project them onto  $\vec{\theta}^{\max}$  along the direction  $\vec{\theta}$  (Fig. 4). The resulting polygons are denoted by  $A_{\vec{\theta}}$  and  $B_{\vec{\theta}}$ . If  $A_{\vec{\theta}} \cap B_{\vec{\theta}} = \emptyset$ , then  $\vec{\theta}$  is

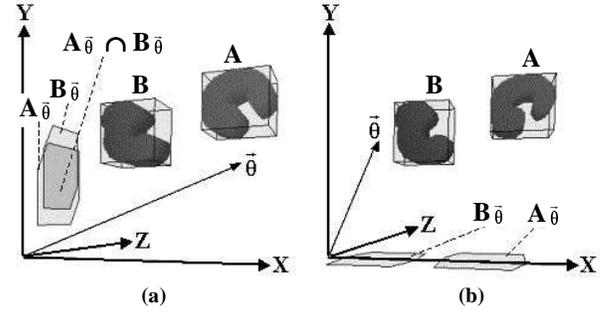


Fig. 4. Relevant directions. (a) The plane  $\vec{\theta}^{\max}$  is the YZ plane, and  $A_{\vec{\theta}} \cap B_{\vec{\theta}} \neq \emptyset$ . The reference direction  $\vec{\theta}$  might be relevant and should be considered. (b)  $\vec{\theta}^{\max}$  is the XZ plane, and  $A_{\vec{\theta}} \cap B_{\vec{\theta}} = \emptyset$ . The direction  $\vec{\theta}$  can be ignored.

irrelevant and should be ignored, otherwise  $\vec{\theta}$  may be relevant and should not be dropped. The intersection of the convex polygons  $A_{\vec{\theta}}$  and  $B_{\vec{\theta}}$  can be determined using, e.g., O’Rourke’s linear algorithm [11].

### 3.3 Relevant Lines

Consider a reference direction  $\vec{\theta}$  that might be relevant according to the test described in Section 3.2. Let  $\Delta$  be an oriented line with direction  $\vec{\theta}$ . This line  $\Delta$  is *relevant* if  $F(A \cap \Delta, B \cap \Delta) \neq 0$ , i.e., if  $A \cap \Delta \neq \emptyset$  and  $B \cap \Delta \neq \emptyset$ . Otherwise, it is *irrelevant*, and need not be considered when computing  $F^{AB}(\vec{\theta})$ . A second test allows us to detect most of the irrelevant lines and, therefore, save additional computational time. Moreover, a set of “canonical” lines that might be relevant—according to this second test—can be rapidly determined and used to assess  $F^{AB}(\vec{\theta})$ .

The intersection  $A_{\vec{\theta}} \cap B_{\vec{\theta}}$  (which is non-empty) is rasterized, i.e., approximated by a set  $R(A_{\vec{\theta}} \cap B_{\vec{\theta}})$  of pixels ( $A_{\vec{\theta}} \cap B_{\vec{\theta}}$  is included in the plane  $\vec{\theta}^{\max}$ , which is either the XY, the XZ or the YZ plane). This is illustrated by Fig. 5. The rasterization can be performed using any standard algorithm (see, e.g., [12]). We have to make sure, however, that the pixels of  $R(A_{\vec{\theta}} \cap B_{\vec{\theta}})$  completely cover the surface  $A_{\vec{\theta}} \cap B_{\vec{\theta}}$ .

Now, for each pixel  $p$  of the plane  $\vec{\theta}^{\max}$ , let  $\Delta_p$  be the oriented line that runs in direction  $\vec{\theta}$  through the center of  $p$ . This line might be relevant if  $p$  belongs to  $R(A_{\vec{\theta}} \cap B_{\vec{\theta}})$ . It is irrelevant otherwise. Let  $\alpha$  be the angle between  $\vec{\theta}^{\max}$  and a plane perpendicular to  $\vec{\theta}$  (Fig. 6). It is easy to show that the value  $[\sum_{p \in R(A_{\vec{\theta}} \cap B_{\vec{\theta}})} F(A \cap \Delta_p, B \cap \Delta_p)] \cos(\alpha)$  is a good approximation of  $F^{AB}(\vec{\theta})$ . Remember that the symbol  $F$  denotes the same numerical function as in Section 2.2. It is defined by a set of algebraic expressions that are determined using integral calculus, and then hard coded [7][8]. At this stage,  $F^{AB}(\vec{\theta})$  can therefore be computed.

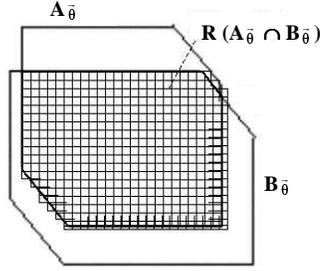


Fig. 5. Rasterization of the intersection  $A_{\bar{\theta}} \cap B_{\bar{\theta}}$ .

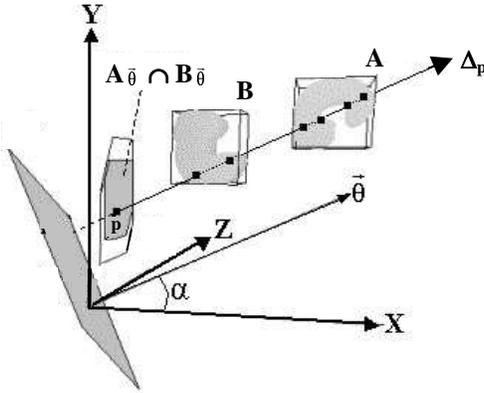


Fig. 6. Relevant lines. Here, the plane  $\bar{\theta}^{\max}$  is the YZ plane. We have:  $A \cap \Delta p \neq \emptyset$  and  $B \cap \Delta p \neq \emptyset$ . Therefore,  $F(A \cap \Delta p, B \cap \Delta p) \neq 0$ , and the oriented line  $\Delta p$  is relevant. The plane represented by the gray parallelepiped is perpendicular to  $\bar{\theta}$ . With  $\bar{\theta}^{\max}$ , it makes the angle  $\alpha$ .

## 4. Experiments

Two series of experiments were performed, on real and synthetic data. The aim of the first series was to compare force histograms with angle histograms in terms of computational efficiency and quality of representation. The aim of the second series was to illustrate a typical application of the histogram of forces.

### 4.1 Force Histograms vs. Angle Histograms

The 3D objects considered in the first series of experiments were two concentric objects: a sphere A and a shell B, as shown by Fig. 7. The values  $\mathcal{A}^{AB}(\bar{\theta})$  and  $F_0^{AB}(\bar{\theta})$ , therefore, were expected not to depend on  $\bar{\theta}$ . The results given in Fig. 8 demonstrate that force histograms preserve space isotropy much better than angle histograms. Note that for bigger objects (i.e., higher values of  $r$ ,  $r_1$  and  $r_2$ ), the force histograms get even flatter.

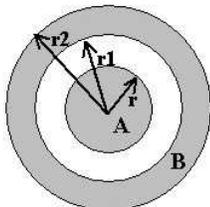


Fig. 7. Concentric objects.

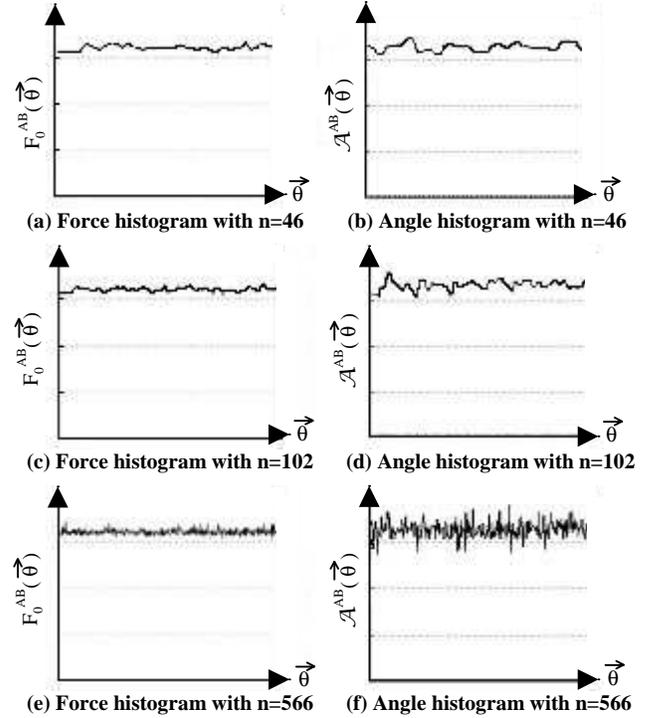


Fig. 8. Isotropy. The  $F_0$  and angle histograms are associated with the pair of concentric objects shown in Fig. 7 ( $r=5$ ,  $r_1=10$ ,  $r_2=16$ ) and computed using (a)(b) 46, (c)(d) 102 and (e)(f) 566 reference directions.

An empirical evaluation of the efficiency of computing force and angle histograms was also conducted. The hardware used was a 2.4GHz Pentium 4 with 512MB of memory. The operating system was Windows 2000 and the implementation language was C. The results presented in Fig. 9a clearly show that force histograms are computed much faster than angle histograms. The main reasons are that not all voxel pairs are considered (a natural selection is induced by the *a priori* digitization of the set of directions in the space, see Fig. 9b) and many voxel pairs are batch processed (through integral calculus and the function  $F$ , as mentioned in Section 2.2). Angle histogram computation, on the other hand, requires all voxel pairs to be considered and processed one by one (the number of reference directions has a negligible impact on the processing time).

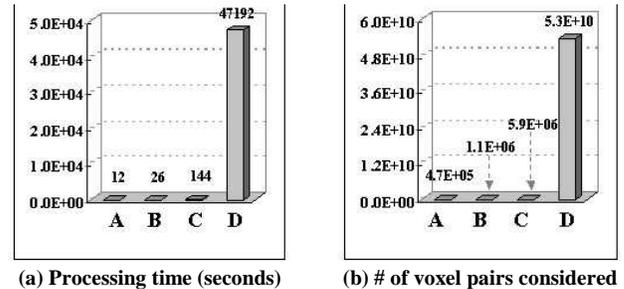


Fig. 9. Computational efficiency. A:  $F_0$ -histogram with forces computed in  $n=46$  reference directions. B:  $F_0$ -histogram with  $n=102$ . C:  $F_0$ -histogram with  $n=566$ . D: angle histogram. All histograms are associated with the pair of concentric objects shown in Fig. 7 ( $r=25$ ,  $r_1=40$ ,  $r_2=64$ ).

## 4.2 A Typical Application

A force histogram associated with a pair (A,B) of 3D objects can be graphically represented by a surface  $[F^{AB}(\vec{\theta}) + r] \vec{\theta}$ . This surface is wrapped around a sphere of arbitrary radius  $r$ , and bumps on it indicate the presence of forces. Figs. 10 and 11 show the histograms of constant forces associated with a pair of synthetic objects and a pair of MRI objects.

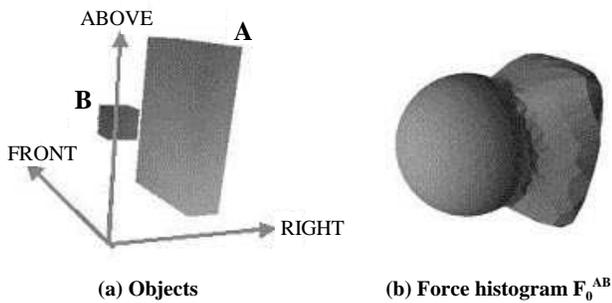


Fig. 10. Force histogram associated with a pair of synthetic objects.

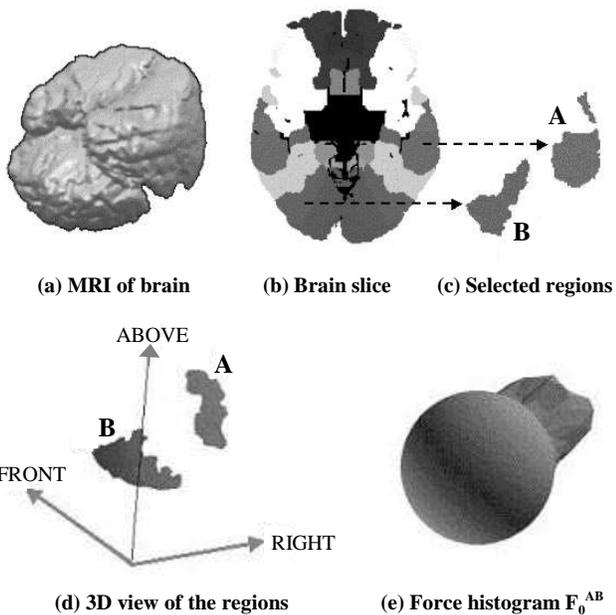


Fig. 11. Force histogram associated with a pair of MRI objects.

Propositions	Synthetic Objects	MRI Objects
A is to the right of B	0.77	0.82
A is to the left of B	0.00	0.00
A is above B	0.16	0.36
A is below B	0.17	0.00
A is in front of B	0.13	0.12
A is behind B	0.13	0.04

Fig. 12. Degrees of truth associated with the six primitive directions for both synthetic (Fig. 10) and MRI (Fig. 11) objects.

The applications of the histogram of forces are numerous [9]. A typical application is the assessment of spatial relationships modeled by fuzzy relations. This is illustrated by Fig. 12, where each value belongs to the interval  $[0;1]$  and corresponds to the degree of truth of some proposition. For instance, 0 indicates that the proposition is completely false, and 1 that it is completely true. All degrees of truth are computed using the method described in [5]. They are computed, however, from the histograms of constant forces associated with the 3D objects, and not, as in the original work, from angle histograms associated with 2D objects.

## 5. Conclusions and Future Work

The histogram of forces is a powerful representation of the relative position between two objects. In previous work, only 2D objects were considered. We have shown in this paper that three-dimensional raster data can be handled as well. The computation of force histograms is much more efficient than the computation of angle histograms. Moreover, space isotropy is much better preserved. In the future, we intend to further extend the notion of the histogram of forces and consider three-dimensional vector data.

## Acknowledgments

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

## References

- [1] J. M. Keller, & X. Wang, Comparison of Spatial Relation Definitions in Computer Vision, *ISUMA-NAFIPS '95*, College Park, MD, 1995, 679-684.
- [2] M. Nabil, J. Shepherd, & Anne H. H. Ngu, 2D Projection Interval Relationships: A Symbolic Representation of Spatial Relationships, *Proc. Of the 4<sup>th</sup> International Symposium on Advances in Spatial Database*, 1995, 292-309.
- [3] J. Li, N. Jing, & M. Sun, Spatial Database Techniques Oriented to Visualization in 3D GIS, *School of Electronic Science and Engineering, National University of Defense Technology*, 2002.
- [4] Eva Stopp, K-Peter Gapp, G. Herzog, T. Laengle, & T. C. Lueth, Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot, *18<sup>th</sup> Annual Conference on Artificial Intelligence*, German, 1994, 39-50.
- [5] R. Krishnapuram, J. Keller, & Y. Ma, Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions, *IEEE Trans. On Fuzzy Systems*, 1(3), 1993, 222-233.

- [6] K. Miyajima, & A. Ralescu, Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations, *Fuzzy Sets and Systems*, 65(2-3), 1994, 225-236.
- [7] P. Matsakis, *Relations spatiales structurelles et interprétation d'images*, PhD Thesis, Institut de Recherche en Informatique de Toulouse, France, 1998.
- [8] P. Matsakis, & L. Wendling, A New Way to Represent the Relative Position of Areal Objects, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 21(7), 1999, 634-643.
- [9] P. Matsakis, Understanding the Spatial Organization of Image Regions by Means of Force Histograms: A Guided Tour, *Applying Soft Computing in Defining Spatial Relations, Studies in Fuzziness and Soft Computing*, 106, 2002, 99-122.
- [10] P. Bourke, Distributing Points on a Sphere, <http://astronomy.swin.edu.au/~pbourke/geometry/>, 1996.
- [11] J. O'Rourke, A New Linear Algorithm for Intersecting Convex Polygons, *Computer Graphics and Image Processing*, 19, 1982, 384-391.
- [12] A. Watt, *3D Computer Graphics*, Third Edition, Addison-Wesley, 2000.