Go to the Right of the Pillar: Modeling Unoccupied Regions for Robot Directives

Marjorie Skubic and Sam Blisard

Dept. of Computer Engineering and Computer Science
University of Missouri-Columbia
Columbia, MO 65211
skubicm@missouri.edu / snbfg8@mizzou.edu

Abstract

It is natural for people to use spatial references in conversation to describe their environment, e.g., "There is a desk in front of me and a doorway behind it" and to issue directives, e.g., "Go around the desk and through the doorway." In this paper, we focus on spatial language to support these directives. We investigate the spatial modeling of regions that do not contain objects but may be referenced by objects in the environment, to compute target destination points for commands such as "Go to the right of the pillar." Two methods are proposed and analyzed using the Histogram of Forces for spatial modeling. We also propose a technique for computing spatial regions which are segmented by confidence level. The paper includes several examples of left, right, front, and rear reference points and spatial regions computed.

Introduction

This work is motivated by the goal of creating a natural, dialog-like interface between a robot and a human user. It is natural for people to use spatial references in conversation to describe their environment, e.g., "There is a desk in front of me and a doorway behind it" and to issue directives, e.g., "Go around the desk and through the doorway." Cognitive models suggest that people use these types of relative spatial concepts to perform day-to-day navigation tasks and other spatial reasoning (Previc 1998, Tversky and Lee 1998, Michon and Denis 2001, Schunn and Harrison 2001), which may explain the importance of spatial language and how it developed. In our research, we have been investigating the use of spatial relationships to establish a natural communication mechanism between people and robots, in particular, striving for an intuitive interface that will be easy for novice users to understand.

In previous work, Skubic *et al.* developed two modes of human-robot communication that utilized spatial relationships. First, using sonar sensors on a mobile robot, a model of the environment was built, and a spatial description of that environment was generated, providing linguistic communication from the robot to the user (Skubic *et al.* 2001a, 2002a). Second, a hand-drawn map was sketched on a Personal Digital Assistant (PDA), as a means of communicating a navigation task to a robot (Skubic *et al.* 2001b, 2002c). The sketch, which represented an approximate map, was analyzed using spatial reasoning, and the navigation task was extracted as a sequence of spatial navigation states. In (Skubic *et al.* 2001c), the results of these two modes were compared for similar, but not exact environments, and found to agree.

In more recent work, robot spatial reasoning is combined with a multi-modal robot interface developed at the Naval Research Laboratory (NRL) (Adams *et al.* 2000, Perzanoswki *et al.* 2001). Spatial information is extracted from an evidence grid map, in which information from multiple sensors is accumulated over time (Martin and Moravec 1996). Probabilities of occupancy are computed for grid cells and used to generate a short-term map. This map is then filtered, processed, and segmented into environment objects (Skubic *et al.* 2002b). Using linguistic spatial terms, a high-level spatial description is generated which describes the overall environment, and a detailed description is also generated for each object. In addition, a class of persistent objects has been created, in which objects are given locations in the map and are assigned labels provided by a user.

The multi-modal interface provides a framework for human-robot interaction. Users can choose among several interface modalities, including speech, gestures, and PDA. Natural language processing and speech synthesis provide the capability of establishing dialogs with the robot. The addition of a spatial language component adds the capability of dialog using spatial references. For example, a user may ask the robot, "How many objects do you see?" The robot responds, "I am sensing 5 objects." The user continues, "What objects do you see?" The robot responds, "There are objects behind me and on my left." More dialog examples are given in (Skubic *et al.* 2002b).

In this paper, we extend our previous work to include spatial modeling of unoccupied regions that are referenced by environment landmarks, for the purpose of issuing directives relative to the landmarks, such as "Go to the right of the pillar." That is, we want to identify a point (or a region) that is to the right of the environment landmark

(the pillar). We will present and compare two methods for computing spatial reference points and analyze the results using regions formed by confidence levels. The method used for spatial modeling is briefly described in Section 2. The methods for computing spatial reference points are presented in Section 3, and analyzed in Section 4. We finish with concluding remarks in Section 5.

Modeling Spatial Relationships

In the context of image analysis, Matsakis and Wendling (1999) introduced the notion of the histogram of forces for modeling spatial relationships between 2D objects. The histogram of forces ensures processing of both raster data and vector data. It offers solid theoretical guarantees and lends itself, with great flexibility, to the definition of fuzzy directional spatial relations such as "to the right of," "in front of," etc. (Matsakis *et al.* 2001). For our purposes, the histogram of forces also allows for a low-computational handling of heading changes in the robot's orientation and makes it easy to switch between an allocentric (world) view and an egocentric (robot) view.

The Histogram of Forces

The relative position of a 2D object A with regard to another object B is represented by a function F^{AB} from $\mathbb{R}$ into $\mathbb{R}_+$. For any direction θ, 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 θ (Fig. 1). F^{AB} is called the *histogram of forces associated with* (A,B) *via* F, or the F–*histogram associated with* (A,B). The object A is the *argument*, and the object B the *referent*. Actually, the letter F denotes a numerical function. Let r be a real number. If the elementary forces are in inverse ratio to d^r, where d represents the distance between the points considered, then F is denoted by F_r. The F_0–histogram (histogram of constant forces) and F_2–histogram (histogram of gravitational forces) have very different and very interesting characteristics. The former coincides with the angle histogram (Miyajima and Ralescu 1994)—without its weaknesses (e.g., requirement for raster data, long processing times, anisotropy)—and provides a global view of the situation. It considers the closest parts and the farthest parts of the objects equally, whereas the F_2–histogram focuses on the closest parts.

Throughout this paper, the referent B is the robot. The F-histogram associated with (A,B) is represented by a limited number of values (i.e., the set of directions θ is made discrete), and the objects A and B are assimilated to polygons and handled through vector data. The computation of F^{AB} is of complexity O(n log(n)), where n denotes the total number of vertices (Matsakis and Wendling 1999). Details on the handling of vector data can also be found in (Skubic *et al.* 2001a, 2002a).

Linguistic Descriptions of Relative Positions

The histogram of forces can be used to build qualitative spatial descriptions that provide a linguistic link to the user. In (Matsakis *et al.* 2001), a system that produces linguistic spatial descriptions of images is presented. The description of the relative position between any 2D objects A and B relies on the sole primitive directional relationships: "to the right of," "above," "to the left of" and "below" (imagine that the objects are drawn on a vertical surface). It is generated from F_0^{AB} (the histogram of constant forces associated with A and B) and F_2^{AB} (the histogram of gravitational forces).

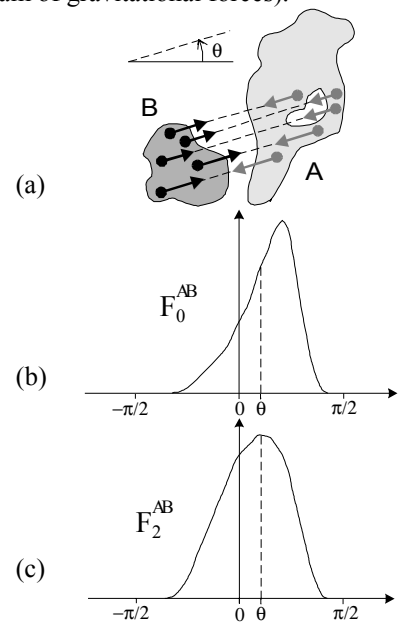


Figure 1. Force histograms. (a) $F^{AB}(\theta)$ is the scalar resultant of forces (black arrows). Each one tends to move B in direction θ. (b) The histogram of constant forces associated with (A,B). It represents the position of A relative to B. (c) The histogram of gravitational forces associated with (A,B). It is another representation of the relative position between A and B.

For any direction θ in which forces are computed, different values can be extracted from the analysis of each histogram. For instance, according to F_r^{AB}, the degree of truth of the proposition "A is in direction θ of B" is $a_r(\theta)$. This value is a real number greater than or equal to 0 (proposition completely false) and less than or equal to 1 (proposition completely true). Moreover, according to F_r^{AB}, the maximum degree of truth that can reasonably be attached to the proposition (say, by another source of information) is $b_r(\theta)$ (which belongs to the interval $[a_r(\theta),1]$). The direction θ for which $a_r(\theta)$ is maximum is called the *main direction*. The main direction is an important feature extracted from the histograms that will be used for determining spatial reference points.

In (Matsakis *et al.* 2001), the "opinion" given by F_r^{AB} about the position of A relative to B is represented by a_r (RIGHT), b_r (RIGHT), a_r (ABOVE), b_r (ABOVE), a_r (LEFT), b_r (LEFT), a_r (BELOW) and b_r (BELOW). Four numeric and two symbolic features result from the combination of F_0^{AB} and F_2^{AB}'s opinions (i.e., of the sixteen corresponding values). They feed a system of fuzzy rules and meta-rules that outputs the expected linguistic description. The system handles a set of adverbs (like "mostly," "perfectly," etc.) that are stored in a dictionary, with other terms, and can be tailored to individual users.

A description is generally composed of three parts. The first part involves the primary direction (e.g., "A is mostly to the right of B"). The second part supplements the description and involves a secondary direction (e.g., "but somewhat above"). The third part indicates to what extent the four primitive directional relationships are suited to describing the relative position of the objects (e.g., "the description is satisfactory"). That is, it indicates to what extent it is necessary to utilize other spatial relations such as "surrounds." When range information is available, a fourth part can also be generated to describe distance (e.g., "A is close to B"), as shown in (Skubic *et al.* 2001a).

Computing Spatial Reference Points

Spatial modeling of the robot's environment is accomplished through the force histograms as described in the previous section, using a vector (boundary) representation for both landmarks and the robot. The map structure used in this work is an evidence grid map (Martin and Moravec 1996). The indoor environment shown in this paper is represented with a 128 x 128 x 1 cell grid, providing a two-dimensional map. One cell covers approximately 11cm x 11 cm. Information from the robot sensors is accumulated over time to calculate probabilities of occupancy for each grid cell. One byte is used to store occupancy probabilities; values range from +127 (high probability of occupancy) to -127 (high probability of no occupancy), with 0 representing an unknown occupancy. For the work reported here, these maps are the sensor-fused short-term maps generated by the robot's regular localization and navigation system (Schultz *et al.* 1999).

The evidence grid map is pre-processed with a sequence of operations, similar to those used for image processing, to segment the map into individual objects. Figure 2 shows an example of a raw grid map with the corresponding segmented environment objects. The contour of each object is used in calculating the force histograms. The robot contour is approximated with a bounding box. For more details and examples, see (Skubic *et al.* 2002b).

The spatial reference points computed are used as target destination points that are to the right and left of an environment object, from the robot's view. In this paper, we consider two methods: (1) Histogram Width and (2) Intersecting Ray. Both methods use features extracted from the force histograms, such as the main direction of the object. The *Histogram Width* method also uses the width of the histogram, as the name implies, to determine left and right points. In the *Intersecting Ray* method, we compute not only left and right points but also points in front and in the rear.

Figure 2. A portion of the grid map. (a) Before pre-processing and segmentation (b) After pre-processing and segmentation. Object #1 corresponds to a row of desks. Object #2 is a file cabinet. Object #3 is a pillar.

The Histogram Width Method

Before computing the spatial reference points, a viewing perspective for the robot is computed, using the main direction of the target object. The main direction is computed as described in Section 2, as the direction that has the highest degree of truth that most of the object lies at that particular angle (with respect to the robot). This should yield a direction that is roughly down the center of the object. Thus, the main direction provides an angular reference of the object's primary direction with respect to the robot. The object's main direction is used to determine the robot's viewing perspective, independent of its actual heading. That is, the spatial reference points are computed as if the robot is facing the object along its main direction.

The force histograms are constructed such that the histogram width indicates the directions in which the robot and the object will both "see" each other. The angles that correspond to the extreme ends of the histogram are called the left and right bounds. Any points outside of this width would show zero forces, so if the robot travels in a path outside of the left and right bounds, it will not collide with the target object. A robot path that aligns perfectly with the left or right bounds would result in the robot just touching the side of the target object.

In the work presented here, we have used features from the constant forces histogram to provide a global view. The difference between the main directions as computed from the two histograms is negligible unless the robot is very close to the object. There is no difference in histogram width between the constant forces and gravitational forces histograms.

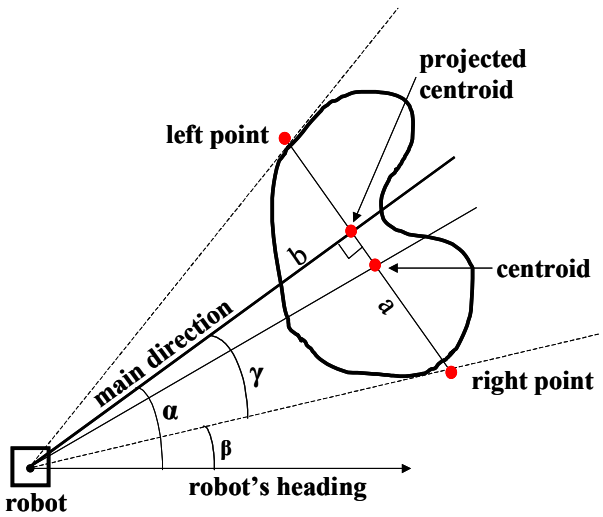


Figure 3. The Histogram Width Method

Figure 3 shows a diagram for the Histogram Width Method. The left and right points are computed such that they lie slightly outside of the left and right bounds of the histogram. (Note that Figure 3 is actually an approximation that assumes the robot is a point. The approximation is valid when the robot is not too close to the object.) The first step is to calculate the centroid of the target object, which is computed as the average of the contour points. Next, the centroid is projected onto the main direction, forming a perpendicular vector as shown in the figure. The left and right points are computed such that they lie on this perpendicular vector. The calculations are straight forward as shown below for the right point:

$$\gamma = \alpha - \beta$$

$$b = \sqrt{(x_{robot} - x_{PC})^2 + (y_{robot} - y_{PC})^2}$$

$$a = b \cdot \tan(\gamma)$$

$$X_R = a \cdot \cos(\gamma - \frac{\pi}{2})$$

$$Y_R = a \cdot \sin(\gamma - \frac{\pi}{2})$$

where α is the main direction, β is the right bound (α and β are both angles referenced by the global frame), b is the distance from the center of the robot to the projected centroid, a is the distance from the projected centroid to the right point, (X_{PC}, Y_{PC}) is the Cartesian coordinate of the projected centroid, and (X_R, Y_R) is the Cartesian coordinate of the right point. The calculations for the left point are similar, except that γ is the angle between the main direction and the left bound.

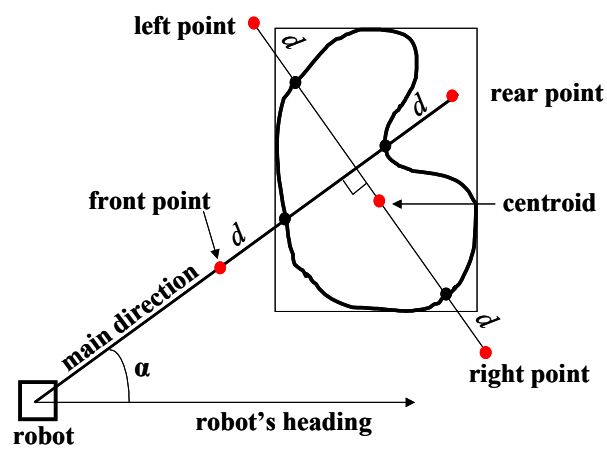
This method provides a strategy for computing points left and right of a target object, along a path that will not collide with the object; however, there are some limitations that will be discussed in Section 4.

The Intersecting Ray Method

This method also considers the main direction as the viewing perspective of the robot. Again, the spatial reference points are computed, as if the robot is facing the target object along the main direction. In this method, reference points will be computed for left, right, front, and rear of a target object, at a specified distance from the object.

Figure 4 shows a diagram for the Intersecting Ray Method. As shown in the figure, a bounding box is constructed by considering the range of (x, y) coordinates that comprise the object contour. The bounding box is used as a convenient starting point for a search of key points along the object contour. This method does not consider the histogram width.

Figure 4. The Intersecting Ray Method



The front and rear points are computed to lie on the main direction vector, at a specified distance, d, from the object boundary. Consider first the front point. Coordinates are calculated along the main direction vector using the following equations:

$$x = r\cos(\alpha)$$
$$y = r\sin(\alpha)$$

where α is the main direction, (x,y) is a point along the main direction, and r is the distance of the vector from the robot to the (x,y) point. Coordinate points are computed incrementally, starting from the robot and checked for intersection with the object contour until the intersection

point is identified. When the intersection point is found, the front point is computed by subtracting the distance, d, from v_F, the vector length of the front intersection point, and computing a new coordinate.

In computing the rear point, we again search for the intersection point of the contour along the main direction vector, this time starting from behind the object. The bounding box of the object is used to compute a starting point for the search. The algorithm first determines the longest possible line through the object by computing l, the diagonal of the bounding box. The starting vector length used in the search is then $v_F + l$. Once the rear contour intersection point is found, the rear point is computed by adding d to the vector length of the rear intersection point and computing a new coordinate.

The left and right points are computed to lie on a vector that is perpendicular to the main direction and intersects the centroid (x_C, y_C) of the object. Again, a search is made to identify the contour point that intersects this perpendicular vector. The starting point for the search of the right intersection point is shown below:

$$x = x_C + l\cos(\alpha - \frac{\pi}{2})$$

$$y = y_C + l\sin(\alpha - \frac{\pi}{2})$$

Once the intersection point is found, a new vector length is computed by adding the distance, d, and computing the new coordinate. The left point is found using a similar strategy to first find the left intersection point.

Analysis of the Two Methods

Both of these methods previously discussed yield points to the left and right of a target object; however they each bring certain advantages and limitations depending upon the situation in which they are employed and the shape of the object.

A simple example of the Histogram Width method is shown in Fig. 5. The primary benefit of the method is that left and right reference points are computed such that a straight line trajectory to the point will not collide with the target object. This is due to the fact that at the left or right bound of the histogram, the robot will not hit the object.

One limitation to this method occurs when the robot is close to an object. As the robot moves closer to the object, the approximation is no longer valid. The histogram width increases, which accentuates the discrepancy, and the left and right point computations yield points far from the robot, as shown in Fig. 6. Fig. 7 shows an example where the left point is so far away that it is off the grid map. In fact, the method provides no way to control the distance from the object.

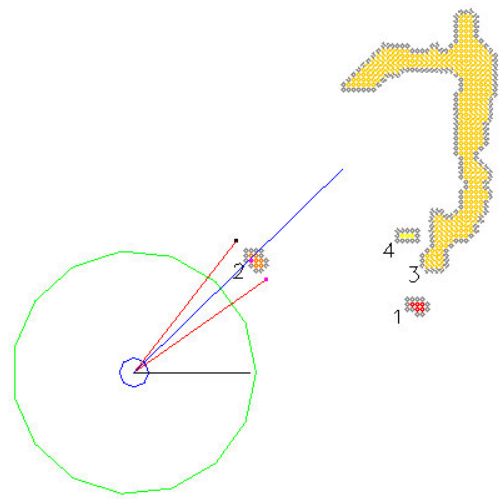


Figure 5. Computing left and right reference points for Object 2 using the Histogram Width Method.

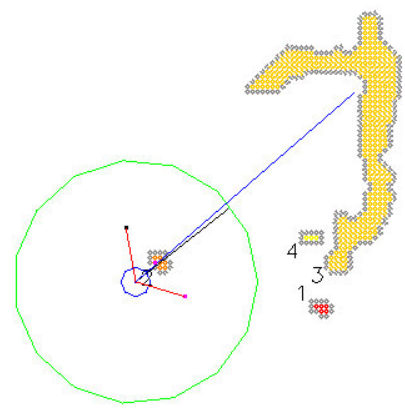


Figure 6. A problem with the Histogram Width Method. As the robot moves closer to Object 2, the histogram width increases.

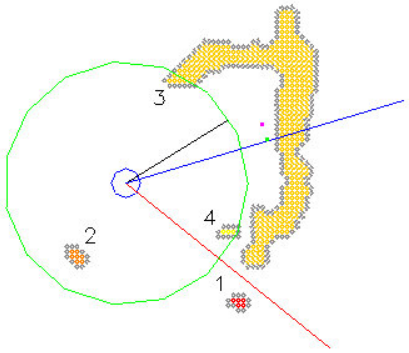


Figure 7. A Bigger Problem with the Histogram Width Method. The left point for Object 3 is off the grid map.

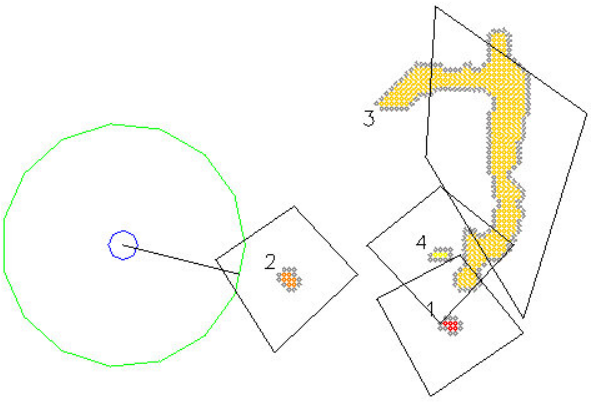


Figure 8. Computing left, right, front, and rear reference points using the Intersecting Ray Method. The vertices of the polygons mark the positions of the left, right, front, and rear points.

The Histogram Width method also fails in a partially surrounded condition, in which case the left and right point computations would be behind the robot. It was this problem that motivated a different way of computing the left and right points.

The Intersecting Ray method solves some of the limitations and can be used to compute all four reference points around the object. Also, the method can be used regardless of the robot-to-object distance and provides a way to specify how far the reference points are from the target object. Fig. 8 shows some examples. The spatial reference points are marked with the diamond polygons around each object; the vertices of the polygons define the left, right, front, and rear points.

Although this method solves some problems, it introduces a new one, at times producing non-intuitive spatial reference points for certain objects. An example is shown in Fig. 9 on the object labeled 3. The problem of non-intuitive points occurs with odd-shaped objects when the robot is positioned such that the rays for perpendicular directions, e.g., behind and left, both intersect the object on the same side. In this case, the resulting points may not appear to be what we, as humans, would naturally consider left and behind. As shown in Fig. 9 (object 3), having both the left and the rear points lie behind the object is non-intuitive at best. Likewise, the right and front points are computed to be in the front, which is also not a natural configuration.

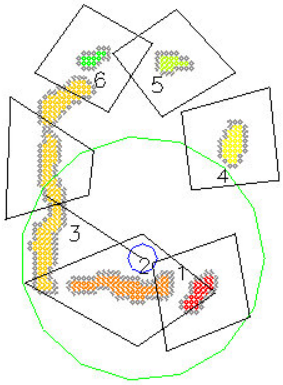


Figure 9. More examples using the Intersecting Ray Method. Note the non-intuitive reference points for Object 3.

In order to resolve these problems, we must determine a method to evaluate the correctness of the left, right, front, and rear points and adjust them as necessary. This evaluation can be accomplished using the same spatial modeling algorithms but switching the argument and the referent (the robot and the object). The degrees of truth are then calculated by placing the robot at the spatial reference points computed. For example, we place a

virtual robot at the left reference point and then run the analysis to determine whether the robot really is to the left of the object. The resulting degree of truth is interpreted as a confidence level. In fact, by placing a virtual robot at neighboring positions, this technique can be used to investigate regions that are to the left, right, front, and rear of an object, where the areas are segmented by confidence level.

Fig. 10 shows an example of regions created using this technique, computed for Object 2. Regions for left, right, front, and rear are shown in the figure (from the robot's perspective). The medium gray regions (green) represent a high confidence level, where the cell i confidence, $c_i \geq .92$. The light gray regions (yellow) have a medium confidence level ($.8 < c_i < .92$). The dark regions (red) have a low confidence level ($c_i \leq .8$).

The figure shows that the regions widen as they become farther from the object, and they join each other in a logical manner. For a relatively small object, the left, right, front, and rear reference points computed by the Intersecting Ray Method lie well within the high confidence region, as shown by the polygon vertices. In the case of the Histogram Width Method, the left and right points computed are also within the high confidence region for the object 2 in Fig. 10, as long as the robot is not very close to the object.

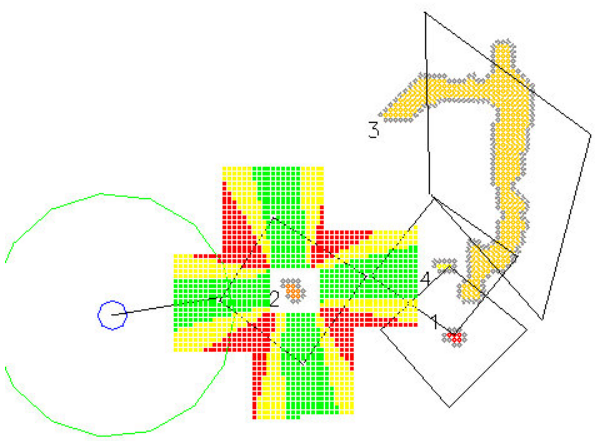


Figure 10. Confidence Regions around Object 2 for left, right, front, and rear spatial references, from the robot's perspective. Medium gray (green) is high confidence. Light gray (yellow) is medium confidence. Dark gray (red) is low confidence.

Fig. 11 shows another example of spatial regions, this time computed for a long object, from a diagonal perspective. Fig. 11a shows the object orientation. Fig. 11b shows the extended regions for the same situation (although the object is no longer visible). Again, the Intersecting Ray Method computes points within the high confidence regions, as long as the specified distance is far enough from the object. Fig. 11b shows the confidence levels of the areas close to the object; it is easy to see that if a close distance is specified, the algorithm could compute reference points in the medium or low confidence regions.

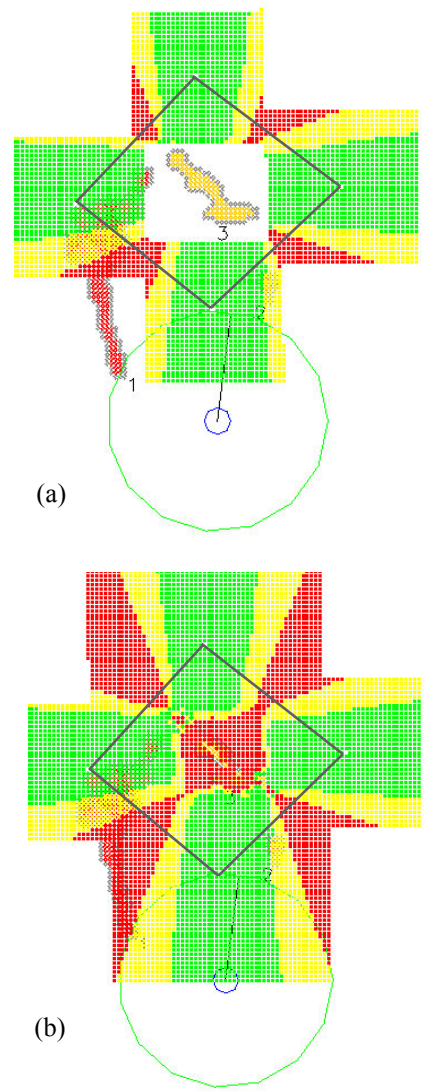


Figure 11. Confidence Regions around Object 3 for left, right, front, and rear spatial references. (a) A partial view showing Object 3 (b) Extended regions.

Finally, let us return to the situation in Fig. 9, where non-intuitive points were computed. This case is repeated in Fig. 12 to show confidence regions for object 3. In this case, the distance from the object must be quite high for all four computed reference points to be in the high confidence region. Even in the example shown, the front and right points are outside of the high confidence area. Thus, we still need a method to establish the validity of a reference point and recompute it if the result is not in a high confidence region.

The computed confidence regions offer a more complete picture of the object's spatial regions, but it is not practical to compute them for each point due to the computational complexity. Our final method will probably employ the Intersecting Ray method to calculate the reference points; then by using those points as starting values, we will perform a hill-climbing algorithm to move the points into the proper region if necessary, with the robot and object switched in the histogram calculations. Hopefully, this final method will allow us to definitively say that the points are in a region considered to be "right," both from a computational and intuitive perspective.

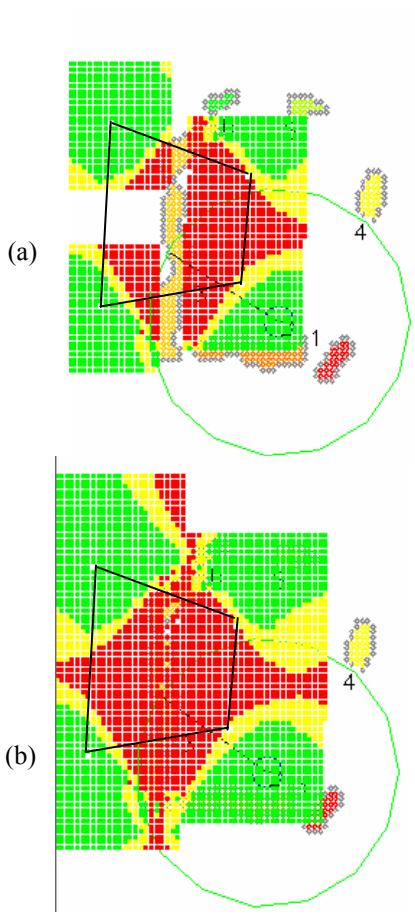


Figure 12. Confidence Regions for Figure 9, Object 3. (a) A partial view showing the object. (b) Extended regions.

Concluding Remarks

In this paper, we have explored two methods for computing spatial reference points relative to environment landmarks. These reference points are intended as destination points in conversation-like robot commands, to allow directives such as "Go to the right of the pillar."

The Histogram Width method is used to compute left and right reference points. The advantage of this method is that the robot can travel along a straight line trajectory to the computed points without colliding with the target object. However, analysis results show that the method is not suitable when the robot is very close to the target object.

The Intersecting Ray method is an improvement, providing a means to compute left, right, front, and rear reference points at a specified distance from the target object. However, further analysis of this method indicates that non-intuitive reference points can be generated in some situations.

We also presented a technique for computing spatial regions based on low, medium, and high confidence levels for the four reference areas. The regions show a larger area that could be used for target destinations, in cases where the previous methods do not compute reasonable points.

Finally, these spatial regions may be useful in other situations, for example, to compute reference points in cluttered environments where a left or right point might coincide with another object in the environment. The varying confidence levels of the regions could also be used to locate more finely specified regions. For example, a user might direct the robot to move "a little to the right" or "somewhat to the right" of the pillar, in which case, a lower confidence level would be acceptable. We intend to pursue this line of investigation in future work.

Acknowledgements

Several MU colleagues and students have contributed to this overall effort, including Professors Jim Keller and Pascal Matsakis and students George Chronis, and Andy Carle. This research has been supported by the Naval Research Laboratory under grant N00173-01-1-G023. The support is gratefully acknowledged.

References

Adams, W., Perzanowski, D., and Schultz, A.C. 2000. "Learning, Storage and Use of Spatial Information in a Robotics Domain", Proc. of the ICML 2000 Workshop on Machine Learning of Spatial Language, Stanford Univ.: AAAI Press, pp. 23-27.

Freeman, J. 1975. "The Modelling of Spatial Relations," *Computer Graphics and Image Processing* (4):156-171.

Keller, J., Matsakis, P. and Skubic, M. 2002. "Beyond 2001: The Linguistic Spatial Odyssey", to appear as a book chapter in *Computational Intelligence Beyond 2001: Real and Imagined*, C. Robinson, ed, Wiley, 2002. Also presented by J. Keller as a Plenary Address, World Congress on Computational Intelligence, Honolulu, Hawaii, May, 2002.

Kuipers, B. 1998. "A Hierarchiy of Qualitative Representations for Space," in *Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, C. Freksa, C. Habel, K. Wender (ed.), Berlin: Springer, pp. 337-350.

Martin, M.C. and Moravec, H.P. 1996. "Robot Evidence Grids", Technical Report #CMU-RI-TR-96-06, Carnegie Mellon University, Mar.

Matsakis, P. 1998. *Relations spatiales structurelles et interprétation d'images*, Ph. D. Thesis, Institut de Recherche en Informatique de Toulouse, France.

Matsakis, P. and Wendling, L. 1999. "A New Way to Represent the Relative Position between Areal Objects", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(7):634-643.

Matsakis, P., Keller, J., Wendling, L., Marjamaa, J. and Sjahputera, O. 2001. "Linguistic Description of Relative Positions in Images", *IEEE Trans. on Systems, Man and Cybernetics*, part B, 31(4):573-588.

Michon, P.-E. and Denis, M. 2001. "When and Why Are Visual Landmarks Used in Giving Directions?" in *Spatial Information Theory: Foundations of Geographic Information Science*, D. Montello (ed.), Berlin: Springer, pp. 292-305.

Miyajima, K. and Ralescu, A. 1994. "Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations," *Fuzzy Sets and Systems*, 65(2/3):225-236.

Perzanowski, D., Schultz, A., Adams, W. and E. Marsh, 1999. "Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy," Proc. of the 1999 IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation, Monterey, CA, Nov, pp.208-213.

Perzanowski, D., Schultz, A., Adams, W., Marsh, E. and Bugajska, M. 2001. "Building a Multimodal Human-Robot Interface," *IEEE Intelligent Systems*, Jan./Feb, pp. 16-20.

Previc, F.H. 1998. "The Neuropsychology of 3-D Space," Psychological Review, 124(2):123-164.

Schultz, A., Adams, W. and Yamauchi, B., 1999. "Integrating Exploration, Localization, Navigation and Planning with a Common Representation," Autonomous Robots, vol.6, no.3, May.

Schunn, C. and Harrison, T. 2001. Personal communication.

Skubic, M., Chronis, G., Matsakis, P. and Keller, J. 2001a. "Generating Linguistic Spatial Descriptions from Sonar Readings Using the Histogram of Forces," in *Proceedings of the IEEE 2001 International Conference on Robotics and Automation*, May, Seoul, Korea, pp. 485-490.

Skubic, M., Matsakis, P., Forrester, B and Chronis, G. 2001b. "Extracting Navigation States from a Hand-Drawn Map," in *Proceedings of the IEEE 2001 International Conference on Robotics and Automation*, May, Seoul, Korea, pp. 259-264.

Skubic, M., Chronis, G., Matsakis, P. and Keller, J. 2001c. "Spatial Relations for Tactical Robot Navigation," in *Proceedings of the SPIE, Unmanned Ground Vehicle Technology III*, April, Orlando, FL.

Skubic, M., Matsakis, P., Chronis, G. and Keller, J. 2002a. "Generating Multi-Level Linguistic Spatial Descriptions from Range Sensor Readings Using the Histogram of Forces," *Autonomous Robots*, Forthcoming.

Skubic, M., Perzanowski, D., Schultz, A. and Adams, W. 2002b. "Using Spatial Language in a Human-Robot Dialog," in *Proceedings of the IEEE 2002 International Conference on Robotics and Automation*, Washington, D.C., May, 2002.

Skubic, M., Blisard, S., Carle, A. and Matsakis, P. 2002c. "Hand-Drawn Maps for Robot Navigation", AAAI 2002 Spring Symposium, Sketch Understanding, Stanford Unversity, March, 2002 (Technical Report SS-02-08).

Tversky, B. and Lee, P. 1998. "How Space Structures Language," in *Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, C. Freksa, C. Habel, K. Wender (ed.), Berlin: Springer, pp. 157-176.