Spatial Relations for Mobile Agent Navigation

George Chronis
Dept. of Computer Engineering and Computer Science,
University of Missouri-Columbia,
email: chronisg@missouri.edu

Abstract

In this paper, we show how linguistic expressions can be generated to describe the spatial relations between a mobile robot and its environment, using readings from a ring of sonar sensors. Our work is motivated by the study of human-robot communication for non-expert users. The eventual goal is to use these linguistic expressions for navigation of the mobile robot in an unknown environment, where the expressions represent the qualitative state of the robot with respect to its environment, in terms that are easily understood by human users. In the paper, we describe the application of the histogram of forces to sonar sensors on a mobile robot. Several environment examples are also included with the generated linguistic descriptions.

1. Introduction

Our work is motivated by the study of human-robot interaction and, in particular, the investigation of human-robot communication. The ultimate goal is to provide easy and intuitive interaction by naïve users, so that they can guide, control, and/or program a robot to perform some purposeful task. We consider the communication between the human user and the robot to be crucial to intuitive interaction by users that are not robotics experts. We further argue that good communications is essential both from the human to the robot (to command the robot to perform purposeful tasks) and also from the robot to the human (so that the user can monitor the robot's current state or condition).

In this paper, we show how linguistic expressions can be generated to describe the spatial relations between a mobile robot and its environment, using readings from a ring of sonar sensors. The eventual goal is to use these linguistic descriptions for navigation of the mobile robot in an unstructured, unknown, and possibly dynamic environment. We are not attempting to build an exact model of the environment, nor to generate a quantitative map. However, we do want to generate linguistic descriptions that represent the qualitative state of the robot with respect to its environment, in terms that are easily understood by human users.

The linguistic spatial descriptions provide a symbolic link between the robot and a human user, thus comprising a navigation language for human-robot interaction. The linguistic expressions can be used for two-way communications with the robot. First, in robot-to-human communication, they provide a qualitative description of the robot's current state (e.g., *there is an object to the left*, or *there is an object to the right front*).

Second, in human-to-robot communication, the human can command the robot to perform navigation behaviors based on the spatial relations (e.g., *while there is an object on the left, move forward*, or *if there is an object on the right front, turn left*, or even a high-level and very human-like directive such as *turn left at the second intersection*). A task can be represented and described as a sequence of qualitative "states" based on spatial relations, each state with a corresponding navigation behavior. We assume the robot has pre-programmed or pre-learned, low-level navigation behaviors that allow it to move safely around its unstructured and dynamic environment without hitting objects.

To accomplish both cases of communication, the robot must be able to recognize its state in terms of egocentric spatial relations between itself and objects in its environment, and it must be able to generate a linguistic description of the spatial relations. The main focus of this paper is the creation of these linguistic spatial descriptions from a ring of sonar sensors.

The idea of using linguistic spatial expressions to communicate with a semi-autonomous robot has been proposed previously. Gribble *et al* use the framework of the Spatial Semantic Hierarchy for an intelligent wheelchair [2]. Perzanowski *et al* use a combination of gestures and linguistic directives such as "go over there" [3]. Shibata *et al* use positional relations to overcome ambiguities in recognition of landmarks [4]. In [5], Stopp *et al* use spatial expressions to communicate with a 2-arm mobile robot performing assembly tasks. Spatial relations are used as a means of identifying an object in a geometric model. That is, the robot has a model of its environment, and the user selects an object from the model using relational spatial expressions.

The work presented here is an extension of spatial analysis previously applied to image analysis. Background material on the spatial analysis algorithms is included in Section 2. In Section 3, we show how the robot's sonar readings can be used to generate inputs for the spatial analysis algorithms. Specific test cases are shown in Section 4 along with a discussion of future work. Concluding remarks are found in Section 5. The interested reader is also referred to a paper on generating linguistic spatial descriptions from sonar readings using the histogram of forces [1] and a paper on using spatial analysis to extract navigation states from a hand-drawn map [6].

2. Background on Spatial Relations

Freeman [7] proposed that the relative position of two objects be described in terms of spatial relationships (such as "above", "surrounds", "includes", etc.). He also proposed that fuzzy relations be used, because "all-or-nothing" standard mathematical relations are clearly not suited to models of spatial relationships. Moreover, "although the human way of reasoning can deal with qualitative information, computational approaches of spatial reasoning and object recognition can benefit from more quantitative measures" [8]. By introducing the notion of the histogram of angles, Miyajima and Ralescu [9] developed the idea that the relative position between two objects can have a representation of its own and can thus be described in terms other than spatial relationships. However, the representation proposed shows several weaknesses (*e.g.*, requirement for raster data, long processing times, anisotropy).

In [10][11], Matsakis and Wendling introduced the histogram of forces. Contrary to the angle histogram, it ensures processing of raster data as well as of vector data. Moreover, it offers solid theoretical guarantees, allows explicit and variable accounting of metric information, and lends itself, with great flexibility, to the definition of fuzzy directional spatial relations (such as "to the right of", "in front of", etc.). For our purposes, the histogram of forces also allows for a low-computational handling of heading changes in the robot's orientation and also makes it easy to switch between a world view and an egocentric robot view.

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 total weight of the arguments that can be found in order to support the proposition "A is in direction θ of B". More precisely, it 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*. Note that throughout this paper, the referent is always the robot. Actually, the letter F denotes a numerical function. Let r be a real. 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—without its weaknesses—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. Details can be found in [1][10][11].

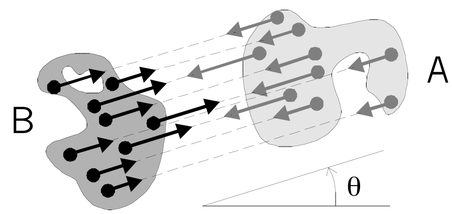

Figure 1. Computation of $F^{AB}(\theta)$. It is the scalar resultant of forces (black arrows). Each one tends to move B in direction θ.

3. Egocentric Spatial Relations from Sonar Readings

In this section, we describe the application of the F_0 and F_2 histograms for extracting spatial relations from the sonar ring of a mobile robot. In our work, we have used a Nomad 200 robot with 16 sonar sensors evenly distributed along its circumference. The sensors' readings are used to build an approximate representation of the objects surrounding the robot. The vertices of each object are extracted and used to build the F_0 and F_2 histograms, which are then used to generate linguistic descriptions of relative positions between the robot and the environment objects (see Figure 2).

The first step in recognizing spatial relations from sonar readings is to build objects around the robot from the sonar readings. Let us consider a simple case of the robot and a single obstacle, shown in Figure 3. The sonar sensor S returns a range value (which is less than the maximum), indicating that an obstacle has been detected. In the case of Figure 3, all sonar sensors except S return the maximum value, which means that no other obstacle was detected. In this case, a single object is plotted as a trapezoid in the center of cone S. The depth of the obstacle cannot be determined from the sonar reading; thus, we use a constant arbitrary depth when building objects. We also represent the cylindrical robot as a rectangular object, because it is easier to process using vector data, since there are only 4 vertices in a rectangle. The bounding rectangle we build around the robot is also shown in Figure 3.

In the case of multiple sonar returns, we examine the sonar readings that are adjacent to each other. There is a question on whether adjacent sonar readings are from a single obstacle or multiple obstacles. Our solution to this issue is to determine if the robot can fit between the points of two adjacent sonar returns. If the robot cannot fit between two returns, then we consider these returns to be from the same object. Even if there are actually two objects, they may be considered as one for robot navigation purposes. In the case that the distance between the two points of the sonar returns is big enough to allow the robot to travel through, we consider separate objects. To form objects from multiple sonar returns we join the centers of the corresponding sonar cones.

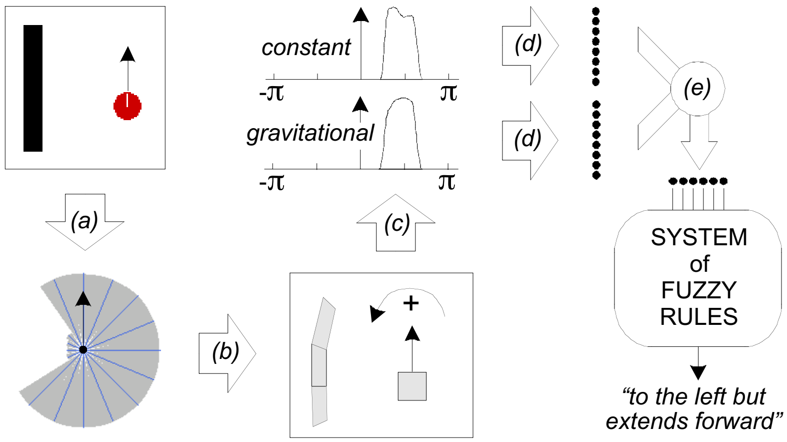

Figure 2. Synoptic diagram. (a) Sonar readings. (b) Construction of the polygonal objects. (c) Computation of the histograms of forces. (d) Extraction of numeric features. (e) Fusion of information.

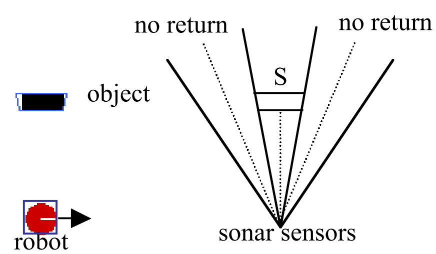

Figure 3. A single object is formed from a single sonar reading.

The distance we compute to determine if two adjacent sonar returns are "close" or not can be expressed by the following formula (distance between two points in polar coordinates):

$$\sqrt{s_1^2 + s_2^2 - 2 s_1 s_2 \cos(2\pi / c)}$$

where: s_1 is the return of sonar sensor S_1,
s_2 is the return of sonar S_2, adjacent to S_1,
c is a constant that determines the angle between the two sonar sensors S_1 and S_2.

For c = 16, the angle between the two sonar sensors is set to the real angle between them ($2\pi/16$), and the formula returns the exact distance between the points of the two sonar returns. However, for our application we used c = 24, for which the distance computed between the points of the adjacent sonar readings is shorter than the actual one.

This way, when the robot diameter is compared to the distance between two obstacles, the distance will be big enough for the robot to easily travel between the obstacles. Thus, we allow extra clearance to make sure that the robot can easily fit between two obstacles.

For example, consider the obstacle in Figure 4. Since the obstacle is relatively far from the robot, the distance between the sonar returns is rather big, and we cannot determine whether the obstacle continues between the three sonar readings, or we have three different obstacles. In this case, we plot three different objects until the robot gets closer to the obstacle and we have a better resolution of the obstacle, since more sensors would detect its presence. In the same figure we show the distance computed for c = 16, which is the distance between A and B, and for c = 24, which is the distance between C and D.

In Figure 5, we show the same obstacle at a closer distance to the robot. There are five adjacent sonar sensors that have returns from the obstacle in this case. The distance measure determines that all sonar returns are close together, for the object to be considered as one.

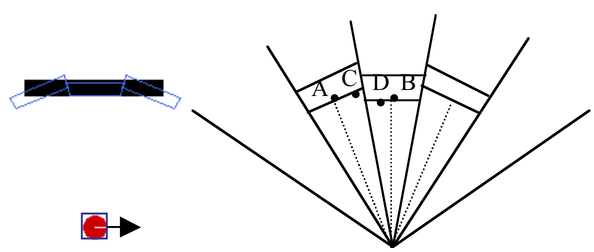

Figure 4. Three different objects are formed from 3 different sonar readings, if the readings are not "close" enough, according to the distance measure.

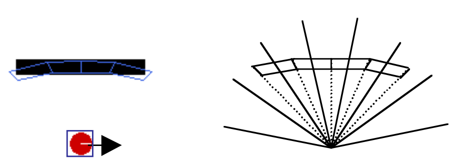

Figure 5. A single object is formed from 5 different sonar readings, if the readings are "close" enough.