Model-Based Echolocation of Environmental Objects*

Juan Carlos Santamaría and Ronald C. Arkin
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
E-mail: {carlos,ashwin}@cc.gatech.edu

Abstract

This paper presents an algorithm that can recognize and localize objects given a model of their contours using only ultrasonic range data. The algorithm exploits a physical model of the ultrasonic beam and combines several readings to extract outline object segments from the environment. It then detects patterns of outline segments that correspond to predefined models of object contours, performing both object recognition and localization. The algorithm is robust since it can account for noise and inaccurate readings as well as efficient since it uses a relaxation technique that can incorporate new data incrementally without recalculating from scratch.

*Technical Report GIT-CS-95/01, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 1995.

1 Introduction

Ultrasonic sensors are relatively inexpensive devices commonly used in mobile robots. These devices calculate range information from the time-of-flight of sonar echoes. Several obstacle avoidance strategies that use this sensor modality have been developed with successful results [1, 4, 7]. These techniques, however, have not dealt with the issue of identifying or recognizing specific object contours in a real world environment.

This paper presents a novel algorithm that can recognize and locate objects using only ultrasonic range data. It exploits a physical model of the ultrasonic beam and combines several readings to extract outline object segments from the environment. The algorithm is then able to detect patterns of outline segments that correspond to predefined models of object contours, yielding object recognition and localization. The algorithm is robust since it can account for both noise and inaccuracies in the data as well as efficient since it uses a relaxation technique that can incorporate new data incrementally without completely recalculating.

This article is divided into the following sections: Section 2 reviews relevant previous work in ultrasonic obstacle localization. Section 3 describes the object recognition and localization algorithm in detail. Section 4 presents qualitative results under simulated and real environments and quantitative results under simulated environments. Section 5 gives some conclusions and outlines for future work.

2 Previous Work

There exists a considerable body of research in the area of obstacle localization using range information provided by an ultrasonic sensor. Several representative examples are discussed below.

Barshan and Kuc [2, 3] describe an efficient, wide-beam sonar system that mimics the sensor configuration of bats using a transmitter (the mouth) flanked by two receivers (the ears). The location of an obstacle is determined by using an unbiased estimator of the time-of-flight (TOF) of the ultrasonic wave. The emphasis of their method is to provide an efficient sensor configuration for obstacle position estimation.

Elfes [8] describes a technique using a data structure referred to as an occupancy grid. This grid represent a sonar map which integrates range measurements from multiple sensors mounted on the robot. The measurements are combined using a method that accounts for the uncertainties and errors inherent in the data. This results in a sonar map that provides for expanding coverage while its definition improves with the addition of new readings. The occupancy grid is used for obstacle avoidance, landmark recognition, and localization (updating the robot's estimate of its position and orientation). This technique, however, is computationally expensive, often requiring several minutes of time on a VAX to perform this match.

Borenstein [5] describes a real-time obstacle avoidance method which uses a histogram grid as a local world model. Range data from ultrasonic sensors are used to continuously update this data structure. A polar histogram is then derived in which each cell corresponds to a sector with a value that corresponds to the obstacle density in that particular direction. Obstacle avoidance consists of moving towards the direction having the least obstacle density. The emphasis is made in detecting occupied areas but with no regard to obstacle shape recognition.

Drumheller [6] developed an algorithm to determine a robot's position and orientation using only ultrasonic range data. The algorithm works by correlating straight segments in the range data against a room model which is described as a list of segments indicating the position of the walls. The algorithm calculates a sonar contour of the room and performs a two-dimensional match (including rotation) between the sonar contour and the room model. A sonar barrier test that exploits physical constraints on the data is used to eliminate implausible configurations. The emphasis is made in contour recognition for robot positioning but not obstacle shape recognition. Also, the algorithm does not integrate measurements over time like the methods that use grids. Thus, it lacks the capability of improving the definition of the contour as the robot gathers more data.

Finally, an algorithm was developed by McKerrow [9] that estimates outline segments using sonar data. The algorithm works by combining two consecutive ultrasonic range readings of the same sensor at two different positions. An outline segment is estimated based on a model of the ultrasonic sensor that consists of an arc of a solid angle equal to twice the beam width. The algorithm calculates outline segments using range data but does not combine the segments together to recognize objects with complex shapes. These features, constructed in this manner, serve as the basis for our research in object recognition.

3 Approach

A novel method has been created for recognizing objects with non-trivial shapes using ultrasonic range data. Unlike the methods described above, our method emphasizes shape recognition as well as obstacle location. It shares in the advantage of the grid-based methods since it improves the definition of the obstacles as more data is added. It also shares in the advantages of the approaches that use models involving physical constraints to reduce the solution space. Unlike other methods, it uses a relaxation technique that significantly improves the reliability of two dimensional matching.

The outline of the algorithm is as follows:

```
For every perception-action cycle:
1.  Read ultrasonic range data.
2.  Extract outline segments.
3.  Cluster outline segments.
4.  Match with shape model.
```

Each of these high-level steps is described in detail in the following subsections.

3.1 Read Ultrasonic Range Data

A Denning sonar ring was the experimental platform used for this research. It has twenty-four laboratory grade Polaroid Ultrasonic Rangefinders equally spaced over 360 degrees in a plane. Range data is obtained by emitting ultrasonic pulses from one or more sensors in a user-specified firing pattern. New sets of sensors are used sequentially by rotating the firing pattern around the ring. This process is repeated continuously as the robot moves through the environment.

A *raw ring reading* consists of an array of range measurements where each cell in the array corresponds to the value of an ultrasonic sensor in the ring. The data reported by the sensors is at a resolution of one-tenth of a foot over a minimum range of one-half of a foot to a maximum distance of 25.5 feet. One raw ring reading is taken at each robot position and stored so it can be processed along with other raw ring readings.

3.2 Extract Outline Segments

Intermediate features are constructed by utilizing two consecutive range readings from a single ultrasonic sensor and combining them to yield an obstacle outline segment. The algorithm described by McKerrow [9] serves as the basis for this method.

This approach exploits a physically-based model of the range readings by creating an obstacle surface segment that is consistent with the characteristics of the sensor readings. A model of the range reading in two dimensions appears in Figure 1. A segment is constructed by calculating the tangent that joins the arcs associated with each range reading. The arc is a sensor model that contains all the points of possible normal surfaces that could produce that reading. Thus, the tangent segment that joins the two arcs is the best estimator of the surface outline that accounts for those readings. The arc is centered by bisecting it with the central axis of the beam. The radius of the arc corresponds to the range measure and the width corresponds to twice the beam angle.

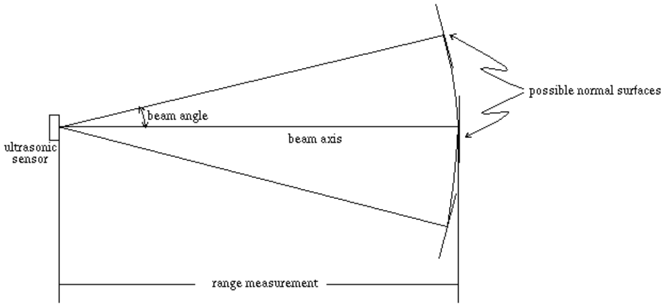

Figure 1: Model of Range Reading

A single measurement does not provide adequate information regarding the obstacle's shape, but two consecutive readings taken at different positions can provide an estimate of the object's outline segment. This estimate consists of a tangent segment joining the two consecutive arcs. The sensor arcs and the resulting estimation of an outline segment is illustrated in Figure 2. For further details about this feature extraction algorithm, the interested reader is referred to [9].

An entire *segment ring reading* consists of an array of outline segments where each cell in the array corresponds to the segment estimated for an ultrasonic sensor in the ring (see Figure 3). One segment ring reading is calculated by combining two consecutive raw ring readings.

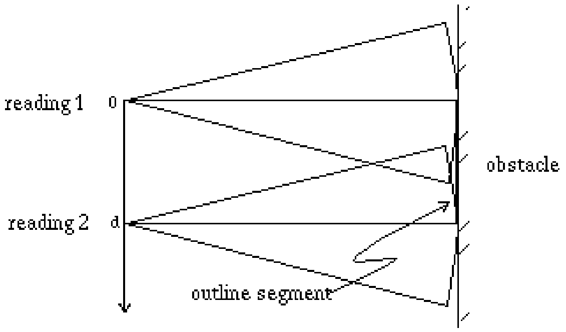

Figure 2: Two Consecutive Ultrasonic Senor Readings

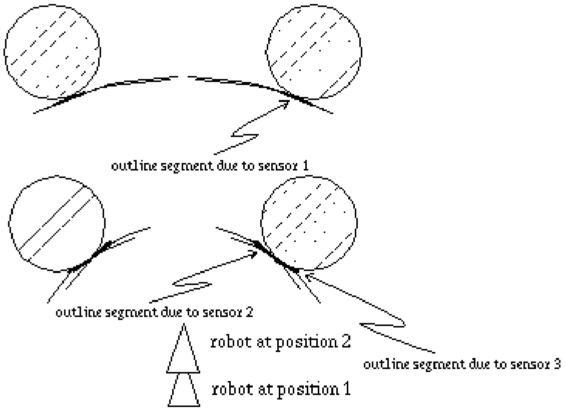
outline segment due to sensor 1
outline segment due to sensor 2
robot at position 2
outline segment due to sensor 3
robot at position 1

Figure 3: Estimated Segment for Ultrasonic Sensor

3.3 Cluster Outline Segments

Using the features extracted by the method described above, a *raw map* of environmental surfaces is obtained by moving the robot while simultaneously calculating and recording multiple segment ring readings. Object shape recognition and localization are performed by analyzing patterns in the raw map. To accomplish this, a simple minimum-distance clustering algorithm that finds and locates clusters of segments is used. These clusters indicate the presence of an object while their centers serve as reliable estimators for the objects' locations.

The minimum-distance clustering algorithm operates over the entire population of segment ring readings. The center and number of segments of each cluster are computed and stored. The center of each cluster provides a good starting point for a more accurate matching process that is explained in the next subsection. Table 1 shows the algorithm in pseudo-code format.

```
Input:      List $\mathbf{S}$ of all available outline segments.
            A threshold distance $T$.
Output:     List $\mathbf{C}$ of clusters.
Algorithm:  1.  Extract segment $S$ from $\mathbf{S}$.
            2.  Create a cluster $C$ with $S$ and add it to $\mathbf{C}$.
            3.  FOR each segment $S$ in $\mathbf{S}$
                    $C_{\mathtt{min}} = \{C_{\mathtt{min}}, C_i \text{ in } \mathbf{C} \mid \|C_{\mathtt{min}} - S\| < \|C_i - S\|\}$
                    IF $\|C_{\mathtt{min}} - S\| > T$
                        Create a cluster $C$ with $S$ and add it to $\mathbf{C}$.
                    ELSE
                        Add $S$ to $C_{\mathtt{min}}$
                        center of $C_{\mathtt{min}} = \overline{\text{center of } S_i}\ \forall S_i$ in $C_{\mathtt{min}}$
```

Table 1: Clustering Algorithm

3.4 Match with Shape Model

In order to recognize an object's shape, the algorithm is supplied with a model consisting of multiple continuous segments (i.e., a *polygonal* approximation). Recognition is accomplished by matching the object model with the segments belonging to a particular cluster and evaluating its goodness of fit. A relaxation method that improves the quality over direct two dimensional matching is employed. It uses a gradient descent technique over an objective function which consists of the sum of the squared orthogonal distances from the outline segments to the model segments. Thus, minimizing the objective function is accomplished by translating and rotating the object model until the best match is found.

To estimate the position of objects in the environment, an object model is placed at each cluster. Then, each object model is updated according to its interactions with the surrounding outline segments. The process stops when the total increment in position and orientation is negligible.

The matching process is now described in more detail. The position and orientation of a candidate object model match is updated by adding small increments to the current position and orientation. These increments are calculated considering pairwise interactions between the segment of the object model and the outline segments estimated from ultrasonic range data. Each outline segment in a cluster serves as an attractor for each segment in the object model (Figure 4) . The interaction between each outline segment and a model segment is such that the candidate model segment moves closer to the data-generated outline segment while at the same time it rotates towards a matching orientation. The total increment applied to the object model is the weighted sum of all the pairwise interactions (Figure 5).

An important characteristic of this approach is that the matching algorithm supports changes in the composition of the raw map of outline segments without drastic influence in performance. Incoming outline segments can be incorporated continuously and old or invalid segments can be removed from the map. The matched object models adjust themselves to take into account the data in the revised map. The adjustment process is quite efficient since only a few iterations are required

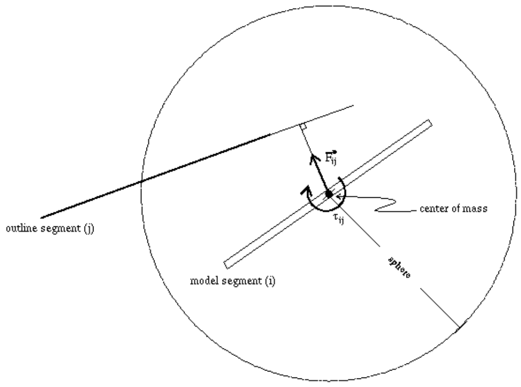

Figure 4: Pairwise Interaction Between a Model and an Outline Segments

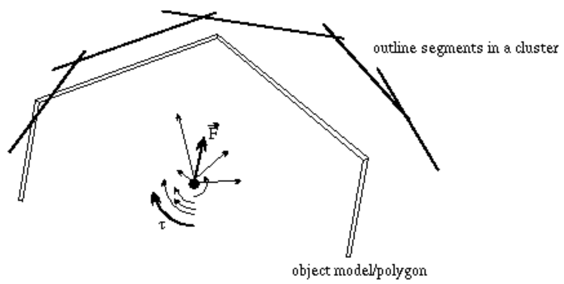

Figure 5: Total Interaction Applied to the Outline Model

to reach the optimum objective if changes in the map are made gracefully and continuously.

Each pairwise interaction between a model segment (i) and an outline segment (j) provides two increments: a translation field ($\vec{F_{ij}}$) and a rotation field (τ_{ij}). Translation is a two-dimensional vector that indicates in which direction the model must be moved and rotation is a scalar that indicates whether the model must be rotated clockwise or counterclockwise. The following formulas are used to calculate their values (Figure 4):

$$\vec{F_{ij}} = \begin{cases} G_t G_r \Delta \vec{N_{ij}} & \text{if } \|\Delta \vec{N_{ij}}\| < sphere_i \\ 0 & \text{otherwise} \end{cases}$$

$$\tau ij = \begin{cases} G_t \Delta \alpha_{ij} & \text{if } \|\Delta \vec{N_{ij}}\| < sphere_i \\ 0 & \text{otherwise} \end{cases}$$

where[1]

$$sphere_i = \gamma \text{ length of model segment } i$$

[1] γ is a constant. We used $\gamma = 2$ in this research.

$$\begin{aligned}
\Delta\vec{N}_{ij} &= \text{perpendicular vector from center of mass} \\
&\quad \text{of model segment } i \text{ to outline segment } j \\
\Delta\alpha_{ij} &= \angle\text{outline segment } j - \angle\text{model segment } i \\
G_t &= \frac{sphere_i - \|\Delta\vec{N}_{ij}\|}{sphere_i} \\
G_r &= \frac{\frac{\pi}{2} - \Delta\alpha}{\frac{\pi}{2}}
\end{aligned}$$

The translation field $\vec{F}_{ij}$ takes the direction of the vector perpendicular to the outline segment $\Delta\vec{N}_{ij}$ from the model's segment center of mass to the outline segment. The magnitude of the translation field vector is computed by G_t and G_r. G_t accounts for the distance between the model segment and the outline segment. The farther the two segments are apart, the smaller the attraction, thus the interaction between the two segments is stronger the closer they are to each other. G_r accounts for the relative orientation between the model segment and outline segment. The more parallel they are the stronger the interaction between them, thus avoiding strong interactions when the model and outline segments are perpendicular.

The total increment in translation and rotation applied to each model segment i is calculated using the following formulas:

$$\begin{aligned}
\vec{F}_i &= \sum_{\text{outline segments } j} \lambda_{ij} F_{ij} \\
\tau_i &= \sum_{\text{outline segments } j} \lambda_{ij} \tau_{ij}
\end{aligned}$$

where

$$\lambda_{ij} = \frac{sphere_i - \|\Delta\vec{N}_{ij}\|}{\sum_{\text{outline segments } j} sphere_i - \|\Delta\vec{N}_{ij}\|}$$

The total increment in translation and rotation applied to each model's segment (i) is a weighted sum of all the interactions of that segment. The weights λ_{ij} corresponds to the relative strength of the interaction ij with respect to all the interactions belonging to the model's segment (i) as indicated in the previous formula. Thus, priority is given to the stronger (i.e., closer) interactions.

The total increment in translation and rotation applied to the candidate model object is calculated using the following formulas:

$$\begin{aligned}
\vec{F} &= \sum_{\text{model segments } i} \mu_i F_i \\
\tau &= \sum_{\text{model segments } i} \mu_i \tau_i
\end{aligned}$$

where

$$\mu_i = \frac{\#\ \text{influencing outline segments}_i}{\sum_{\text{model segments } i} \#\ \text{influencing segments}_i}$$

Similarly as with the interactions of each model's segment, the increment in translation and rotation applied to the whole candidate object model is the weighted sum of the individual model's segment increments. Each weight μ_i corresponds to the relative strength of model's segment (i) with respect to the rest of the model's segments. The larger the number of interactions (i.e., influencing outline segments) the higher the weight.

4 Results

The algorithm described in this paper has been run on both simulated and real sonar data. A graphical tool running under Openwindows 3.0 has been developed to aid in visualizing and manipulating the sonar data. The same tool can also be used to control either a real or simulated robot. The remainder of this Section describes some results obtained from various experiments using these methods.

4.1 Experimental Settings

An objective was to be able to detect and localize 55 gallon steel drums in a relatively unstructured environment. To that end, a simulated environment was created containing six drums, each with a 2 foot diameter, arranged in two columns separated by 6 feet. Within each column, the drums had a constant separation of 1 foot. A real world environment was also tested by using 8 drums arranged in two columns. The drums, however, were not placed regularly within each column to show that the algorithm is able to detect the drums even in irregular configurations. A Denning MRV-III mobile robot was used for gathering data. The simulator is based on this robot as well. The mobile robot has a diameter of 2.5 feet.

For both the simulations and the actual robotic experiments, the robot navigated using schema-based reactive control [1] with **move-ahead** and **avoid-static-obstacle** behaviors active.

A model based on a semi-circle was provided to the algorithm for both environments. The semi-circle was approximated by five segments of equal length with a 2 foot diameter.

Figures 6, 7, 8, and 9 show the output of each of the phases of the algorithm for the simulated environment and Figures 10, 11, 12, and 13 do the same for the real environment.

4.2 Discussion

In the simulated environment, there is no noise nor invalid ultrasonic range readings. Nevertheless, it is still a difficult task to extract useful object outlines using single ultrasonic readings (Fig. 6). A better recovery of the environment is obtained by imposing a physical model for the range readings

Drum	World coordinates (ft.,ft.)	Estimated world coordinates (ft.,ft.)
Drum 1	(1.00, 4.00)	(1.21, 3.81)
Drum 2	(4.00, 4.00)	(4.02, 3.83)
Drum 3	(7.00, 4.00)	(6.80, 3.86)
Drum 4	(1.00,-4.00)	(1.32,-3.86)
Drum 5	(4.00,-4.00)	(4.11,-3.82)
Drum 6	(7.00,-4.00)	(6.84,-3.83)

Table 2: Simulation Results

and combining them to eliminate implausible solutions (Fig. 7). A simple clustering algorithm then selects groups of segments that require further processing to extract relevant information (Figure 8 shows the clusters as rectangles with an "X" indicating the center of the clusters). Finally, the relaxation algorithm progressively updates the position and orientation of candidate object models until the best match is found (Figure 9).

Due to the nature of the ultrasonic sensors, a considerable number of readings from different perspectives must be combined to allow precise location of objects in the environment. As can be seen in Figure 9, the only way the robot can precisely detect the location of the lower drums is only after traversing the hallway far enough to allow the constructed outlines segments to form a semi-circle in the raw map. The situation is not the same for upper drums where the outline segments form only a quarter of a circle. Even then, however, the relaxation algorithm is able to find the best estimation of the location of the upper drums given the current level of information. Moreover, as more outline segments are included in the raw map, the better the estimate for the upper drum's location becomes.

The performance of the algorithm in a real environment is presented in the next set of figures to show that it can cope with noise and inaccurate range readings (Fig. 10). The drums were placed slightly moved in and out along the hallway to show the reliability of the algorithm in detecting their position. Figure 11 shows the raw map consisting of outline segments, Figure 12 shows the position of clusters, and Figure 13 shows the best estimation of the location of each drum.

As in the simulated environment, the number of readings influences the confidence of the location of the objects. As the robot gathers more readings from different perspectives the better the raw map becomes and the better the algorithm estimates the location of the objects in the environment. The algorithm is able to accomplish this efficiently since new information is incorporated gracefully and continuously due to the nature of the navigation task which does not permit large spatial discontinuities in the world (i.e., the sonar readings are gathered frequently during the robot's path traversal).

Table 4.2 shows the actual world coordinates for the drums and the estimated world coordinates obtained with the algorithm for the simulated environment.

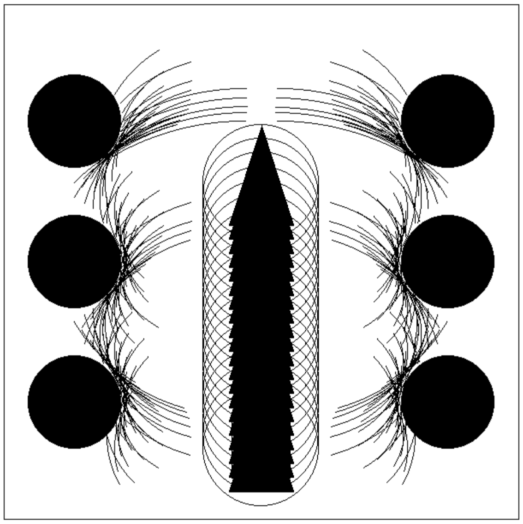

Figure 6: Simulation - Raw Data

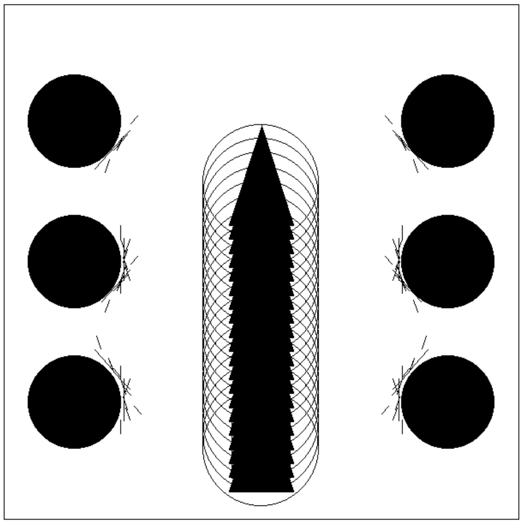

Figure 7: Simulation - Segments

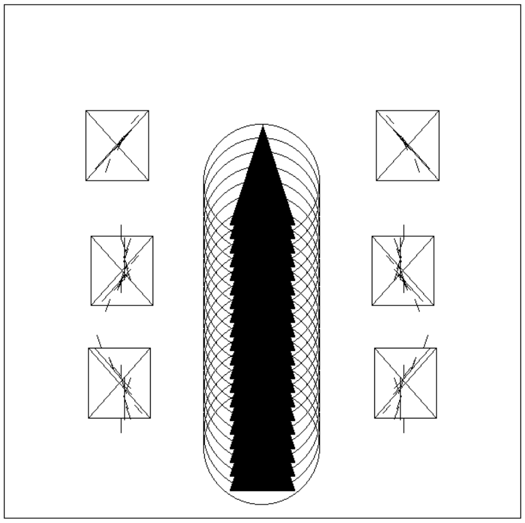

Figure 8: Simulation - Clusters

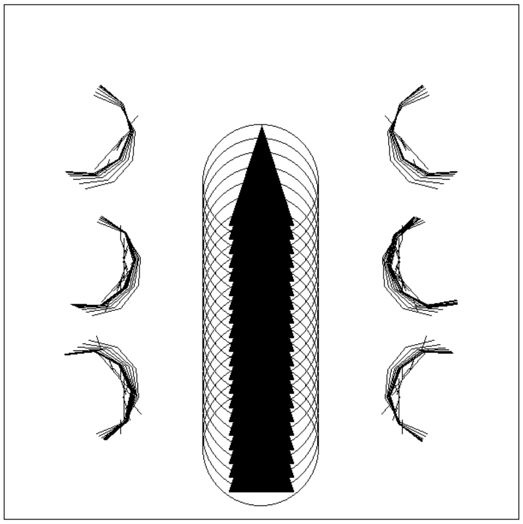

Figure 9: Simulation - Drums

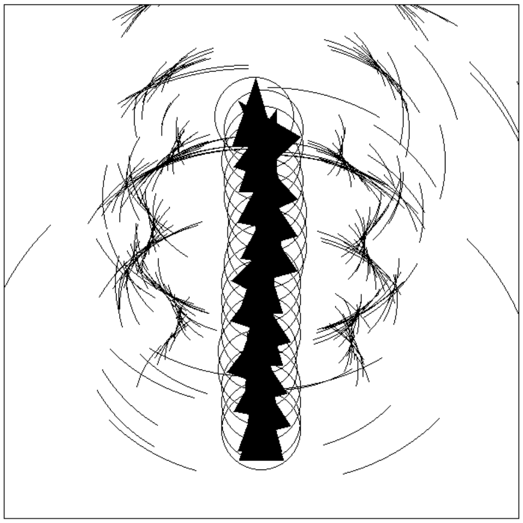

Figure 10: Real Run - Raw Data

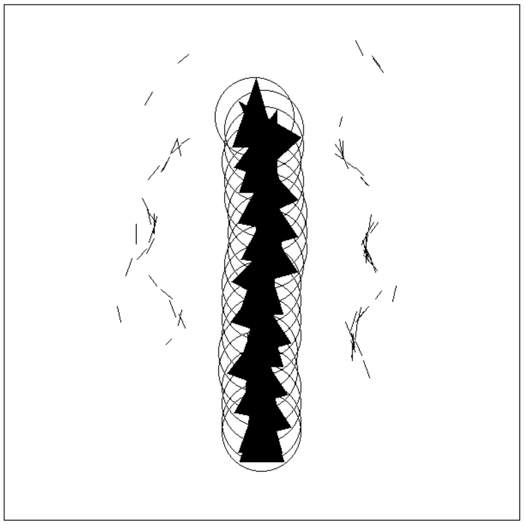

Figure 11: Real Run - Segments

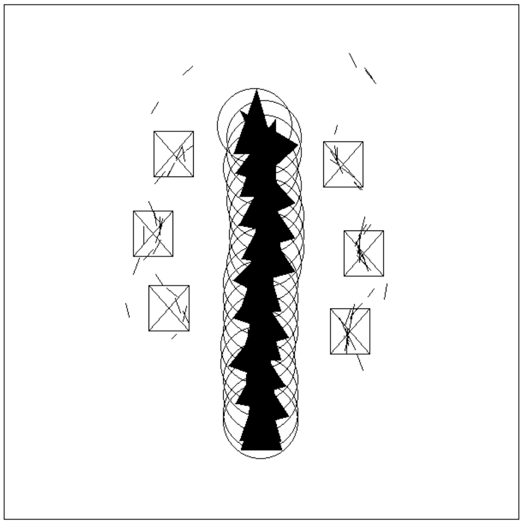

Figure 12: Real Run - Clusters

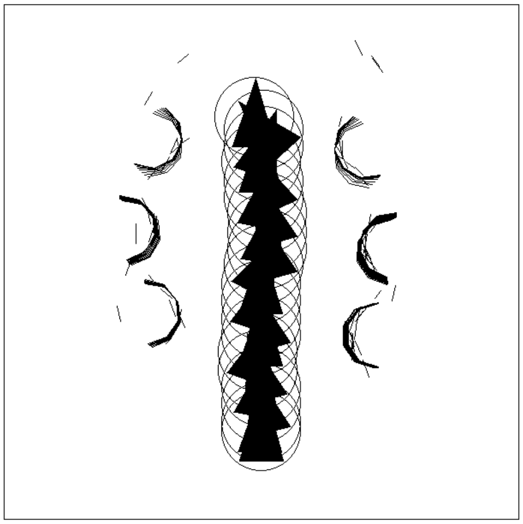

Figure 13: Real Run - Drums

5 Summary and Conclusions

An algorithm that is able to recognize the shape and location of objects in the environment using only range ultrasonic data was presented. The algorithm exploits the physical model of the ultrasonic beam and knowledge of the shape of the objects to accomplish its task. Outline segments are extracted from the raw range readings to form a map of the environment that consists of object surface outlines. Then, the outline segments are clustered to find possible candidates for object locations. Finally, a relaxation algorithm progressively refines the match between the shape of the object and the outline segments by incrementally moving and rotating the object shape model until the best match is found.

The algorithm is efficient since it avoids standard two dimensional matrix matching to find patterns in the raw map. Instead, it uses pairwise interactions between intermediate features (i.e., outline segments) and model segments to evaluate and improve the match. This significantly reduces the data that must be processed resulting in improvement of the run-time efficiency while maintaining the accuracy of the matching process.

The algorithm shares the advantages of grid methods by allowing incremental incorporation of new data. Incoming information can be added to the raw map and the location and orientation of the detected objects can be updated in an efficient manner. The algorithm also shares the advantages of model-based methods by exploiting physical constraints to discard alternatives and reduce the solution space.

Future research involves the application of the algorithm in robot localization and more sophisticated object recognition. For localization, the algorithm has the capability to recognize and estimate the relative position of several objects. This may be used in structured environments to update the robot's location by comparing the relative position of the drum columns as the robot moves.

For sophisticated object recognition, the algorithm can be extended to recognize patterns of objects such as box canyons which can constitute a problem for purely reactive robotic platforms. Once detected, the box canyon's outline can influence the decision of what action to take in conjunction with other sensory information perceived by the platform. This can significantly enhance the performance of the reactive system.

Acknowledgments

This research was supported by the Westinghouse-Savannah River Technology Center. Additional funding for the Mobile Robotics Laboratory is provided by the National Science Foundation under grants #IRI-9100149 and #IRI-9113747 and ARPA under ONR Grant #N00014-94-1.

References

[1] R.C. Arkin, "Motor Schema Based Mobile Robot Navigation", *International J. of Robotics Research*, 8, 4, pp. 99, (1989a).

[2] B. Barshan and R. Kuc, "A Bat-Like Sonar System for Obstacle Localization", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, 4, 1992.

[3] B. Barshan and R. Kuc, "ROBAT: A Sonar-Based Mobile Robot for Bat-Like Prey Capture", *Proc. of the 1992 IEEE International Conference on Robotics and Automation*, pp. 274–279, 1992.

[4] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots", *IEEE Transactions on Systems, Man, and Cybernetics*, vol 19, 5, pp. 1179–1187, 1989.

[5] J. Borenstein and Y. Koren, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions on Robotics and Automation*, vol 7, 3, pp. 278–288, 1991.

[6] M. Drumheller, "Mobile Robot Localization Using Sonar", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, 2, pp. 235–332, 1987.

[7] A. Elfes, "Sonar-Based Real-World Mapping and Navigation", *IEEE Journal of Robotics and Automation*, vol. RA-3, 3, pp. 249–265, 1987.

[8] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", *Computer*, vol. 22, 6, pp. 46–57, 1989.

[9] P.J. McKerrow, "Echolocation: From Range to Outline Segments", *Proc. International Conference on Intelligent Autonomous Systems*, pp. 239–247, Pittsburg, PA, 1993.