Obstacle Avoidance and Path Planning Using a Sparse Array of Sonars

Matei Ciocarlie
Computer Science Department, Columbia University
New York, NY 10027, U.S.A.

Abstract
This paper proposes an exploration method for robots equipped with a set of sonar sensors that does not allow for complete coverage of the robot's close surroundings. In such cases, there is a high risk of collision with possible undetected obstacles. The proposed method, adapted for use in urban outdoors environments, minimizes such risks while guiding the robot towards a predefined target location. During the process, a compact and accurate representation of the environment can be obtained.

1. Introduction

The problem of autonomous obstacle avoidance in unknown, or partially known environments using only sonar sensors for perception has been extensively studied by using robots well adapted to sonar sensing, often equipped with a complete ring of sonars, covering as many as 16 sensing directions around the robot. These approaches may be unsuitable for large robots equipped with relatively few sonars.

The most important shortcomings of the sonar sensor are well known: inaccuracy in establishing the azimuth location of a detected obstacle and non-detection of obstacles intersected by the sonar beam at an angle far enough from perpendicular. The usage of a small number of sensors on a large robot greatly emphasizes the latter: at any given time, only few sensing directions are covered, increasing the possibility for undetected obstacles. In this document, we will discuss the case of a large robot equipped with a sparse sonar array, covering only 6 sensing directions. We propose an exploration method that minimizes the risk of collision by taking advantage of the characteristics of a man made outdoor environment.

2. Previous work

Due to its low cost, the sonar is a popular sensor for autonomous robot navigation and the study of its use has lead to numerous results ranging from modeling the response of a single sensor or combining multiple readings for improved azimuth accuracy to using probabilistic techniques for mapping and localization in complex environments.

A large fraction of the work on modeling sonar range returns concentrates on improving the azimuth accuracy of the readings [2][4][8]. There are multiple reasons for this focus. First, the testing environment is usually indoors, of small size and cluttered with small obstacles, in which case very precise mapping of features such as edges and corners is required. Second, non-readings due to angled reflections are not a factor due to a complete sensing array around the robot.

The issue of using a sparse array of sonars in a large environment is substantially different. The importance of azimuth accuracy is strongly diminished, since it is used for correcting errors that are one order of magnitude smaller than the size of the robot and up to two orders of magnitude smaller than the total size of the obstacle. As mentioned before, while the importance of azimuth accuracy is diminished, the importance of non-detection due to wide angles is strongly increased.

Another way of interpreting range data, closely related to the method presented in this paper is the extraction of lines from sets of 2-D points. McKerrow proposes in [1] a technique that uses any two sonar readings along the same direction whose arcs accept a common tangent. For our exploration method, we do not rely on the fact that two displaced readings of the same obstacle will be available at any given time, and displacing the robot for obtaining another reading cannot be safely done until the orientation of the obstacle has been detected. Furthermore, using just two sonar readings, this method is susceptible to strong influence by noise. Other proposed methods of line extraction from range data are more robust, using more available readings [5][6]. It is important to note that most of these results are subsequently used to solve the localization problem in indoors environment. The goal is therefore to extract very accurate lines using range data, and the sensor of choice is almost always a laser range finder, providing higher accuracy and dependability than the sonar. Our method uses line fitting to range data for the more immediate goal of creating a collision-free path around an obstacle.

As simultaneous mapping and localization is a highly active research area, a number of successful and robust methods have been found; many of them are probabilistic in nature and do not rely on explicit feature extraction from range data. For a survey and classification of these methods, we refer the reader to [7]. Most often, experimental robots make use of a full ring of sonars. Furthermore, the mapping and localization problem is often separated from exploration, which is the situation in which the robot needs to make its own decision on choosing movement directions. The uncertainty introduced by the lack of reliable data on the immediate surroundings prevents using these techniques and necessitates focus on the lower level problem of minimizing the risk of collision.

3. Detecting and matching obstacle orientation

As mentioned before, the first priority of this method is deciding on a movement direction that minimizes chances of collision. What causes this problem is the fact that few sensors are available to monitor the area directly in front of the robot, therefore this area is sensed under a very limited number of different orientations. Further complicating the problem is the fact that planar surfaces, such as walls, are abundant in a man-made environment, and cannot be detected if the angle of the beam is not close to perpendicular on the orientation of the wall.

The situation described in figure 1.a. assumes a sonar array oriented under 5 different angles, three of them pointing in front of the robot and one to each side. Possible obstacle orientations are shown, differentiated by sonar detection: green obstacles can be detected by the front sonars, blue ones are only seen by a side sonar while red ones are completely undetected. We infer two results: the number of totally undetected obstacles is relatively small, but not insignificant, while information gained from the side sonar needs to be used to make assumptions about possible obstacles located in front of the robot.

Another problematic situation is depicted in fig 1.b. The robot detects an obstacle directly in front and tries to take an avoiding direction. The most obvious method involves turning until front sensors show a clear path and advancing, which will almost certainly result in a collision.

The solution is detecting the exact orientation of the wall and planning a course that is parallel to it. In the case of obstacles detected only by a side sonar, determining its exact orientation and matching it will eliminate the risk of frontal collision (figure 1.c). Furthermore, it will permit maintaining a strong contact with the wall through the side sonar whose beam is now perpendicular on it. In the case of a frontal obstacle, an avoiding path that is parallel to it will again be the safest one (figure 1.d.). The use of a side sonar that is perpendicular on the wall will give a good reading on when it is safe to “turn the corner”.

The case of possible undetected obstacles (depicted in red in figure 1.a.) requires further details. While this method cannot completely eliminate this risk, it

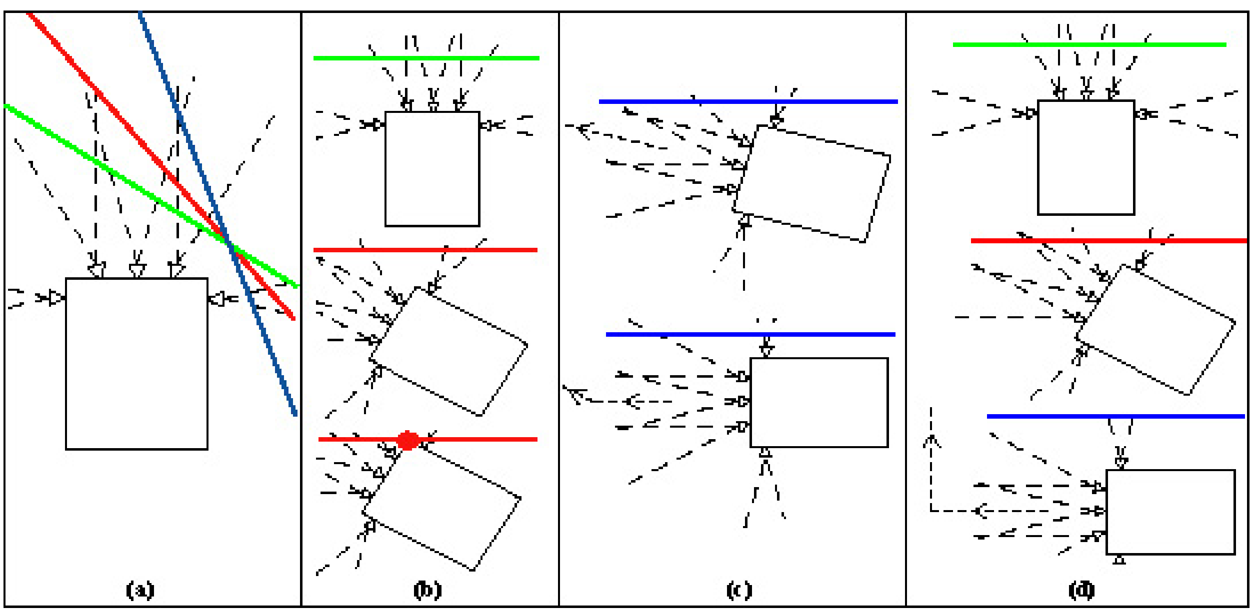

Figure 1. (a) Possible obstacle orientations in relation to the sonar sensor; the green wall is detected by the front sonars, the blue wall is detected only by the side sonar, the red wall is not detected. (b) Naïve avoidance method results in collision as robot turns until no sensor can detect the wall. (c) Side sonar is used to correct a path that would lead to collision with a wall not detected by a front sonar. (d) Correct avoidance method: when an obstacle is detected the robot plans a path that is parallel to the obstacle

substantially lowers it taking advantage of the environment characteristics. Most man-made structures are practically entirely composed of perpendicular or parallel outer walls. If the direction of movement is parallel to a wall, it is very likely that a next possible obstacle will be perpendicular on it, increasing chances of detection. The general rule that guides this method is that whenever an obstacle is detected, a path that is parallel to the orientation of the obstacle must be chosen. This will not only minimize the chance of frontal collision but also allow for a strong contact with the obstacle using side sonars.

The method used for determining the orientation of an obstacle consists of using the Hough transform for fitting a line to sonar returns obtained while the robot is turning to achieve a parallel orientation. As the robot turns, more readings are obtained and the orientation of the obstacle is continuously refined. The precision of the final result is in most cases within 5 degrees. It is important to note that a larger error, sometimes possible due to the nature of the wall or imprecise sonar returns will not likely lead to a collision, as it will be in most cases detected and corrected by a side sonar, whose beam is now in contact with the wall.

4. Mapping and Path Planning

Using the method discussed above, we now focus on the greater goal, of planning a path to a final destination. With this purpose, a map of all previously detected obstacles must be maintained and path planning function must make use of all the information.

4.1 Obstacle representation

In our representation, numerous line segments are used to approximate any obstacle. More precisely, whenever the robot will sense an obstacle, it will accurately detect its orientation as well as distance from the robot. It will then place on the map a segment characterized by that particular position and orientation. The length of a segment is arbitrary, since the robot has no way of knowing the span of the obstacle, at that time. A compromise value must be used, since longer segments will imply worse approximations of actual obstacles as well as over conservative paths around them, while shorter segments will necessitate stopping for measurements more often.

Representing all obstacles by line segments has obvious limitations, since irregular or round shapes will not be closely approximated. In a man-made outdoors environment however, planar and almost planar surfaces are predominant. Small irregularly shaped obstacles will also be approximated as planar surfaces, but the only drawback is that a more conservative path around will be planned, a minor drawback compared to the greater task of avoiding walls and buildings.

As opposed to the popular occupancy grid model, a segment-based map is extremely compact and well suited for use in large environments. An occupancy grid approach involves a tradeoff: large cells imply loss of resolution, already limited by usage of sonars, while smaller cells are computationally expensive to handle in the case of large maps. The segment representation allows the usage of maximum resolution of the sonar sensor, while being compact enough for representing very large regions.

4.2 Path planning

The goal of the path planning function is to re-compute a safe path to a goal whenever the obstacle map is updated. Our exploration method prevents the use of efficient grid-based re-planning algorithms, such as D* ([3]). The reason is the robot needs to be allowed to plan a path that is parallel to an obstacle, while grid-based algorithms limit possible movement direction to neighboring cells. Using framed quadtrees, described in [9], instead of a conventional uniform grid allows for many angles of direction in unoccupied areas, while in the vicinity of obstacles movement is still restricted.

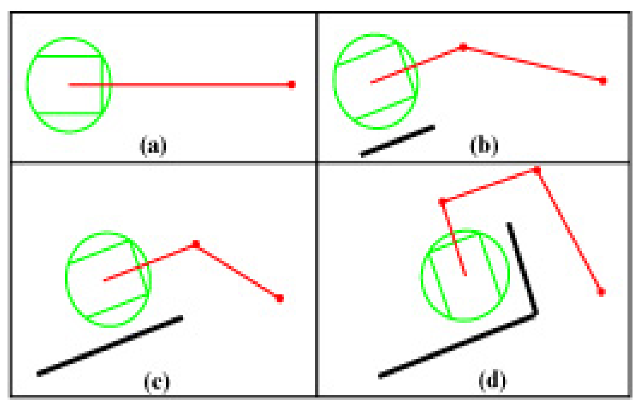

Figure 2. (a) Initial path to goal. (b) Side sonar detects an obstacle; a new path is generated in order to avoid frontal collision. (c) Side obstacle is still detected, a new segment is added to its representation. (d) A frontal obstacle is now detected, waypoints are added in order to avoid it

Since a path that is parallel to an obstacle is often required, we model the chosen path in a similar way to obstacles, through line segments. The path is defined by a number of waypoints and any new obstacle (or extension of a previously known obstacle) is handled by adding new waypoints that take the robot around it. Whenever a new obstacle segment is placed on the map, the path planner will decide whether it is to be avoided by turning left or right and adds waypoints accordingly. The result is depicted in figure 2. while further implementation details can be found in B.

5. Experimental results

Experimental test runs have been conducted in an outdoors environment dominated by man-made structures such as buildings, pillars and fences. The obtained results are depicted in figures 3 and 4. For each run, figure a. shows a 2-dimensional map of the test area, as well as the start and goal points chosen for the robot. Figure b. shows the obstacles detected by the robot during exploration as well as the traveled path.

In both cases the goal point was set in a way that would require extensive exploration of the building facades. The robot correctly re-planned the path to the goal while moving along walls, without colliding with any obstacle. Smaller obstacles such as mailboxes and air conditioning units (shown in the 2-D map of the area) were also correctly avoided. The tests showed that our method is indeed reliable for exploration in such environments and is able to handle large obstacles such as buildings with a very low risk of collision.

6. Further possibilities – localization through mapping

The general advantages of line-based map representation are summarized in [6]. Replacing point maps by line segments is a highly effective form of compression. Further compression is possible by merging similar segments, using a method like the one shown in [6]. For structured environments, segments form a more accurate representation of the actual obstacles and noisy and inaccurate readings are also filtered out by this method..

All of these characteristics can be found in obstacles maps generated through our exploration method. This is a direct result of the features of the outdoors environment, as large buildings dominate the landscape. In the case of an incorrect line fit leading to a robot orientation that is not parallel to the wall, the estimation is corrected as soon as side sonars discover the error, and the presence of a small number of incorrectly oriented segments does not heavily effect the representation of the whole building. In the case of a correct estimate, the pose is maintained along the entire length of the wall, since a correct parallel path is planned, leading to a correct representation.

The result, in the absence of accumulating odometry errors, is the one seen in figure 4b. It represents a compact, yet accurate representation of the region. A mapping heavily influenced by odometry errors can be seen in figure 3b. As we have mentioned, parallel and perpendicular lines are predominant in the chosen environment type, making the accumulation of odometry error detectable through the presence of angles close but not equal to 0 or 90 degrees. Future work will include using such results for pose correction or global localization.

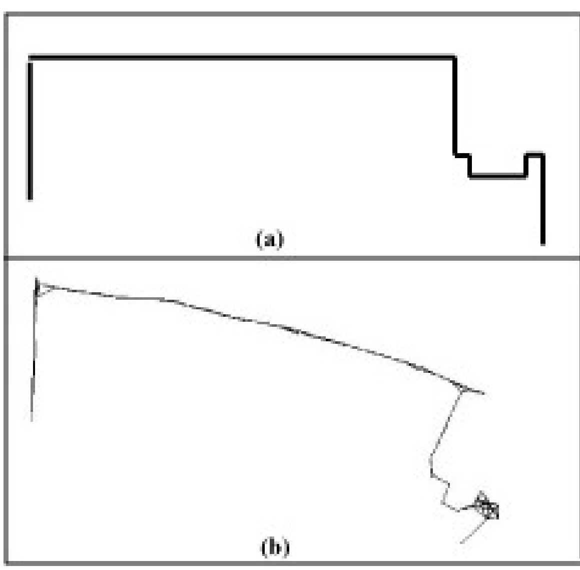

Figure 3.

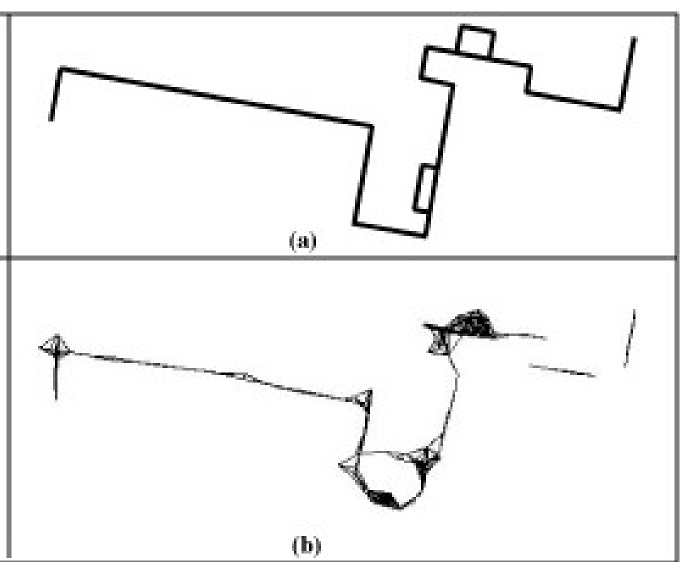

Figure 4.

7. Conclusion

We have presented an exploration method for large robots sparsely equipped with sonar sensors operating in a man shaped outdoors environment, dominated by planar surfaces such as walls or fences. By continuously approximating the orientation of encountered obstacles and planning a path that is parallel to them, our method minimizes the risk of collision with undetected obstacles. During the process a compact mapping of the encountered obstacles is obtained and can be applied to correcting pose errors or solving the global localization problem.

A. Using the Hough transform to detect obstacle orientation

The proposed algorithm for detecting an obstacle's orientation assumes approximating multiple sonar returns with a line. The final result is the position (distance from the robot) and orientation of the obstacle. The method used for approximation is the Hough transform.

While the method described in this paper focuses on situations where little data is available on the immediate surroundings, the line-fitting algorithm requires multiple readings on the same obstacle. This data is obtained while the robot is turning in order to avoid the obstacle, motion that is necessary regardless of the used algorithm. As the robot turns, new data is continuously added, refining the estimate of the obstacle orientation. One of the main advantages of the Hough transform is that improving an estimate by adding new data is not computationally expensive as it does not involve a recomputation "starting from scratch": the new data is simply added to the Hough space and a new maximum is found.

The usage of the Hough transform also has shortcomings. This method will find the line that best fits as many points as possible and all other points will have no effect on the final result; this behavior can lead to imperfect results for our exploration method. We have chosen to use it mainly because of the computational advantage described above and for its simplicity – due the nature of our exploration method better precision is in most cases not required as incorrect fits are corrected while the robot follows the path around the obstacle. The usage of other line extraction methods, such as the ones shown in [5] and [6], can also be explored.

B. Recursive path-planner for obstacles approximated by segments

The simple path planner previously described might fail in cases where a path around the newly discovered obstacle can not be found because of previously discovered walls. In such a case the path planner will look for a way around by going around many segments in a recursive fashion:

1. If path intersects segment
   1. Increase path length
   2. Check if segment can be avoided on left side
   3. If new path intersects another segment
      1. Call path planner for path on the left side
      2. Value returned is length of path to the left
   4. Check if segment can be avoided on right side
   5. If new path intersects another segment
      1. Call path planner for path on the right side
      2. Value returned is length of path to the right
   6. Return shortest path (left or right)
2. Else return

This recursive path planner will eventually search the entire tree of possible paths and return the path that goes around the smallest number of obstacles (not necessarily the shortest in Euclidian distances). While the path planner is very stable and will return a correct path in most encountered cases, the depth of the search has been artificially limited, in order to avoid infinite searches. For this reason the path planner might fail when trying to find the way out of a bottleneck in a very cluttered map. Improving search speed and allowing for deeper searches will alleviate the problem but not eliminate it. Further refinements of this method should therefore also focus on improving the path planning function.

References

[1] P. J. McKerrow, "Echolocation – from range to outline segments", *Robot Autonomous Systems*, 1993

[2] O. Wijk and H Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking", *IEEE Trans. Robot Automat.*, vol. 16, pp. 740 – 752, 2000

[3] A. Stentz, "Optimal and Efficient Path Planning for Partially Known Environments", in *Proc. of the IEEE Int. Conf. On Robotics and Automation*, May 1994

[4] H Choset, K. Nagatani and N. A. Lazar, "The Arc-Transversal Median Algorithm: A Geometric Approach to Increasing Ultrasonic Sensor Azimuth Accuracy", *IEEE Trans. Robot Automat.*, vol. 19, pp. 513 – 522, 2003

[5] G. A. Borges and M.J. Aldon, "A split-and-merge segmentation algorithm fir line extraction in 2-d range images," in *Proc. 15th Int. Conf. On Pattern Recognition*, Barcelona, Sept. 2000

[6] S.T. Pfister, S .I. Roumeliotis, J. W. Burdick, "Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation"

[7] S. Thrun, "Robotic Mapping: A Survey", *Exploring Artificial Intelligence in the New Millennium*, Morgan Kauffman, 2002

[8] J. J. Leonard and H. F. Durrant – White, *Directed Sonar Sensing for Mobile Robot Navigation*. Norwell, Ma: Kluwer, 1992.

[9] A. Yahja, S. Singh, A. Stentz, "Recent Results in Path Planning for Mobile Robots Operating in Vast Outdoor Environments", in Proc. *1998 Symposium on Image, Signal Processing and Robotics*, Sept. 1998