

Arc Carving: Obtaining Accurate, Low Latency Maps from Ultrasonic Range Sensors

David Silver, Deryck Morales, Ioannis Rekleitis, Brad Lisien, and Howie Choset
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract—In this paper we present a new technique for improving the azimuth resolution of ultrasonic range sensors frequently used with mobile robots. This improvement is achieved without a significant increase in the latency, or processing delay, of the system. Our approach decreases the azimuth uncertainty of a sensor reading by eliminating portions of the reading that are contradicted by subsequent readings. Our idea bears resemblance to space carving as used by the vision community, where a ray of light is used to define the boundaries of an obstacle. A sonar model similar to that commonly utilized by occupancy grids is used. Our method, termed arc carving, can be used to produce maps that are both accurate and with low enough latency for robust mobile robot navigation. Experimental results verify this approach over spaces as large as 5000 square meters.

1. INTRODUCTION

This work originated during experiments in Simultaneous Localization and Mapping (SLAM), previously described in [1]. A mobile robot is tasked with fully exploring an unknown space while correcting for positioning error. Range sensors serve three purposes in these experiments: detection of obstacles for navigation and obstacle avoidance, determination of the boundary of the space to ensure complete coverage, and feature extraction to aid in localization. To this end, readings from the ultrasonic sensors are processed to produce a two dimensional local map. This map consists of points believed to be on the boundary of freespace.

During these experiments, an important tradeoff was encountered between the accuracy and latency of the local map. In this context, latency refers to the time between receiving information from a sonar sensor, and points corresponding to this information appearing in a local map. Low latency sonar processing techniques, such as assuming the point of reflection lies directly in front of the sensor, have high azimuth uncertainty that results in an inaccurate local map. Inaccurate local maps can lead to failure of the localization and coverage tasks of the robot. In contrast, techniques that provide higher azimuth resolution, such as the Arc Transversal Median (ATM)[2] approach, require fusing multiple sonar readings, resulting in a higher latency local map. High latency local maps can cause the robot to fail coverage and obstacle avoidance, due to the absence of newly detected obstacles in the local map.

Our solution to this problem is a new technique, which we call *arc carving* after its similarity to space carving [3] used in the vision community. Arc carving fuses multiple sonar readings to improve azimuth resolution, but in a serial manner that still produces low latency local maps. Arc carving can

also be used in conjunction with existing sonar processing techniques.

In the next section, background and previous work in sonar processing are discussed. Section 3 presents a description of the arc carving technique. Improvements offered by arc carving are described in Section 4. Section 5 presents experimental results, and we conclude with a discussion and directions for future work.

2. PREVIOUS WORK

Ultrasonic range finders use time of flight to measure distance. The sensor returns a range r , and the angle of the sensor relative to the robot, θ , is known. However, this range is not the straight line distance to an obstacle along θ , but rather the distance to the point of reflection of the obstacle [4]. This point could be anywhere along the perimeter of the sensor's beam pattern (Fig. 1(a)). The region of highest response of this pattern can be sufficiently approximated by an arc of radius r centered at orientation θ . For the Polaroid transducers used in our work, the arc width, denoted by ϕ , is 22.5 degrees. In contrast, on many mobile bases the discretization in range of these sensors is one inch. The final result is a sensor fairly accurate in depth, but not in azimuth [5]. Therefore, improving sonar accuracy involves improving azimuth resolution.

A. Centerline Model

A simplistic model of sonar behavior is the centerline model. The centerline model assumes that the point of reflection is located at the center of the sonar arc. The advantage of the centerline model is its simplicity—it does not require taking multiple readings to produce range data that can be

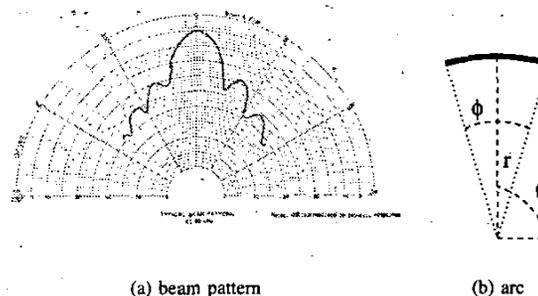


Fig. 1. Beam pattern for the Polaroid transducer installed on many mobile robots, and an arc approximation of the main lobe

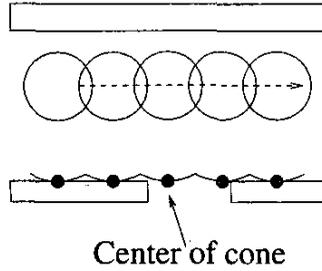


Fig. 2. The centerline model often fails to detect doorways and narrow corridors

incorporated into a local map. The disadvantage is that it has poor accuracy, because the actual point of reflection could be anywhere along the arc. The most striking example of this problem is the case where the robot fails to detect a doorway or other narrow opening, [6], as in Fig. 2.

B. Occupancy Grids

A well-known approach to sonar processing is the occupancy grid [7] [8] [9] [10]. The world is divided into a two dimensional grid, with each cell containing two values, representing occupancy status and certainty. A common sonar model for occupancy grids is a planar cone, where each point in the cone is part of a probability distribution indicating the likelihood that there is an obstacle at that point (Fig. 3). The probability of occupancy is high along the base of the cone, and near zero in the interior. This is a key contribution of the occupancy grid: it interprets from a sonar return not only a region where an obstacle is likely to be, but also a region where it is unlikely for an obstacle to be. The grid cells are updated by combining these probabilities of occupancy using Bayes Rule.

The disadvantage of occupancy grids is that they require a tradeoff between resolution and computational resources. The higher the desired resolution, the more finely discretized the grid must become, resulting in larger demands on memory and computation. Since sonar readings must be processed in real time while sharing resources with navigation and localization software, this is a serious consideration. Conversely, as the size of the environment grows, the discretization must become coarser in order for the demands on memory and computation to remain constant. Given that we are operating in environments on the order of thousands of square meters, the granularity required for efficient use of occupancy grids is a concern.

C. Line Fitting Techniques

Several techniques have been developed that work by fusing multiple sonar readings to try and gain a more abstract understanding of obstacles and their locations. In [11] McKerrow fits line segments that are tangent to multiple sonar arcs. Leonard et al. group sonar arcs that have similar depth as coming from the same obstacle [12]. And MacKenzie and Dudek fit line segments to clusters of sonar points. [13]. While these techniques all provide information suitable for robot navigation,

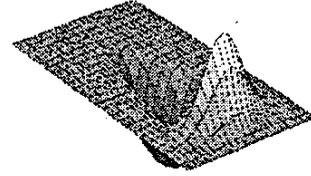


Fig. 3. Gaussian Distribution along the arc of a sonar cone

they are not always accurate enough for localization. Also, they all suffer from the same narrow opening problems that the centerline model does. Finally, requiring multiple points to fit lines results in an increase in latency.

D. Arc Transversal Median (ATM)

The ATM [2] method uses an arc model similar to the centerline method. Each sonar return has an origin (x, y) corresponding to the location of the sensor, a range r , a sensor orientation θ and a beam width ϕ . The probability of a point on the arc perimeter being the point of reflection is modeled as a uniform distribution over the curve

$$C = \begin{bmatrix} x + r \cos(\tau) \\ y + r \sin(\tau) \end{bmatrix} \quad \left(\theta - \frac{\phi}{2} \leq \tau \leq \left(\theta + \frac{\phi}{2} \right) \right) \quad (2.1)$$

and zero elsewhere. If two curves from different sonar arcs intersect each other in a transversal manner, then the region of intersection is stable and may be used as a better estimate of the point of reflection.

A history of the most recent arcs is kept. Whenever a new arc is added, it is checked for intersections with all other arcs in the history. If an intersection exists and meets a minimum transversal angle threshold, the intersection is stored. When a local map of the processed range data is requested, ATM

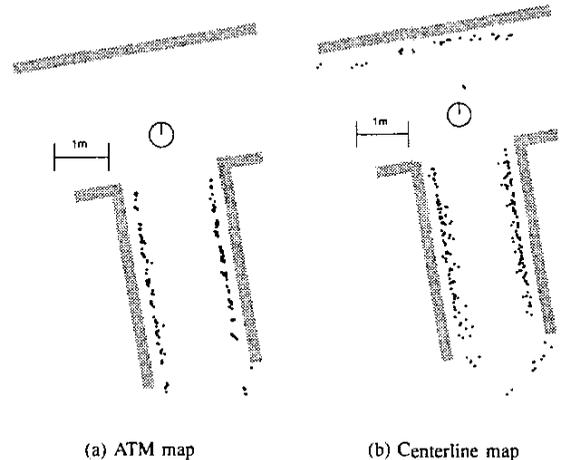


Fig. 4. Local maps produced by the ATM and Centerline methods as the robot approaches a 'T' junction of two hallways. Points in the ATM map correspond to the medians of transversal intersections, points in the Centerline map correspond to the center of arcs. Light grey obstacles have been drawn in for the sake of display. The ATM map does not yet contain any points corresponding to the approaching wall, demonstrating the problems associated with high latency.

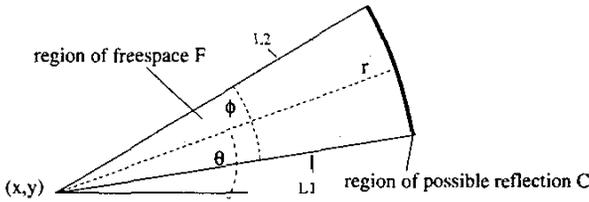


Fig. 5. The regions of freespace and possible reflection for a sonar arc

returns the median of all stored intersection points along each arc. If there are no intersections along an arc when a map is requested, ATM defaults to using the centerline model for that particular arc.

This approach provides a drastic improvement in azimuth resolution, and also eliminates the narrow opening problem. However, it also has considerable latency problems. In order to obtain at least one intersection with a minimum transversal angle ψ , two arcs must have origins that differ by a minimum distance D_{min} , defined by

$$D_{min} = 2r \sin\left(\frac{\psi}{2}\right) \quad (2.2)$$

Thus, the farther away the obstacle is, the farther the robot must travel before it can obtain transversal intersections. This often results in intersections not occurring until well after the robot has already passed by the obstacle they correspond to. In the interim, the robot is navigating using a map that is generated mostly by the centerline model. During this period, the robot may fail its coverage or navigation task due to an inaccurate local map. So while the centerline method provides data that is timely but not accurate enough for navigation and localization, ATM provides data that is accurate but not timely enough. Fig. 4 shows an example of this problem. While ATM generates a much cleaner map (Fig. 4(a)), it has not yet generated any data off the wall the robot is approaching, giving the robot a vastly different view of the world than the centerline model (Fig. 4(b)). In practice, the ATM and centerline maps would be combined to produce a more complete, but inaccurate local map.

3. ARC CARVING

The basic premise of arc carving is that a sonar return actually provides two pieces of information instead of one: it indicates a region where the sonar reflection could have occurred, and another region that is likely to be freespace (Fig. 5). These regions correspond to the regions of high and low obstacle probability of an occupancy grid sonar model [7]. When the region of possible reflection from one sonar return intersects the region of freespace of another sonar return, then these two readings represent inconsistent information. Since these two arcs might have been taken from radically different locations in both time and space, the more recent of the two is considered authoritative. If the freespace region of a newer arc overlaps the region of possible reflection of an older arc, then the older arc is updated to be consistent (Fig. 6). In this way the region of possible reflection is carved away. When requested by the navigation software, a local map containing

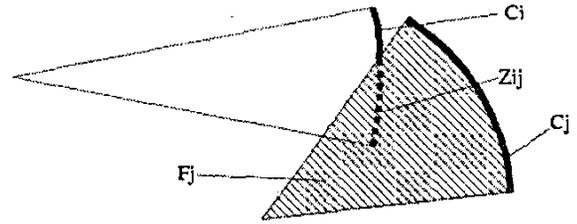


Fig. 6. The portion of a region of possible reflection that is overlapped by a region of freespace is removed

one point per carved arc is returned. As these arcs decrease in size, so does the maximum error of the corresponding point in the local map.

Although arc carving involves fusing multiple sonar readings, it is still a low latency approach. Since there are no thresholds for the quality of data that need be met, points are added to the local range map as soon as an obstacle is detected. The accuracy of points corresponding to the obstacle is then improved upon as more sonar readings are taken from the obstacle. Arcs are combined in a serial manner, so there is no waiting for a minimum number of arcs.

Arc carving uses a sonar model similar to ATM. More formally, for each sonar arc a 5-tuple is stored, consisting of an origin (x, y) , a range r , and a range of angles $[\theta_1, \theta_2]$. This range is initialized as

$$[\theta_1, \theta_2] = \left[\theta - \frac{\phi}{2}, \theta + \frac{\phi}{2}\right]$$

where θ and ϕ are the sensor orientation and sonar beam width, respectively.

The region of possible reflection is the arc defined as

$$C = \left[\begin{array}{l} x + r \cos(\tau) \\ y + r \sin(\tau) \end{array} \right] \quad \theta_1 \leq \tau \leq \theta_2 \quad (3.1)$$

The freespace region of a reading, F , is bounded on one side by C , and on two sides by the line segments L_1 and L_2 from the origin of the arc to the endpoints of C (Fig. 5). F could also be defined by

$$F = \{(x', y') | \sqrt{(x' - x)^2 + (y' - y)^2} \leq r, \theta_1 \leq \arctan\left(\frac{y' - y}{x' - x}\right) \leq \theta_2\} \quad (3.2)$$

In our mapping scheme, the local range map consists of one point P_i for every arc. If C_i is connected, P_i is defined as

$$P_i = \left[\begin{array}{l} x + r \cos\left(\frac{\theta_1 + \theta_2}{2}\right) \\ y + r \sin\left(\frac{\theta_1 + \theta_2}{2}\right) \end{array} \right] \quad (3.3)$$

In the case where $[\theta_1, \theta_2] = \left[\theta - \frac{\phi}{2}, \theta + \frac{\phi}{2}\right]$, which we term the uncarved case, the point returned is identical to that of the centerline model.

Given two sonar readings i and j

$$Z_{ij} = C_i \cap F_j, \quad (3.4)$$

is the set of points which are in the region of possible reflection of reading i and the region of freespace of reading j (Fig. 6).

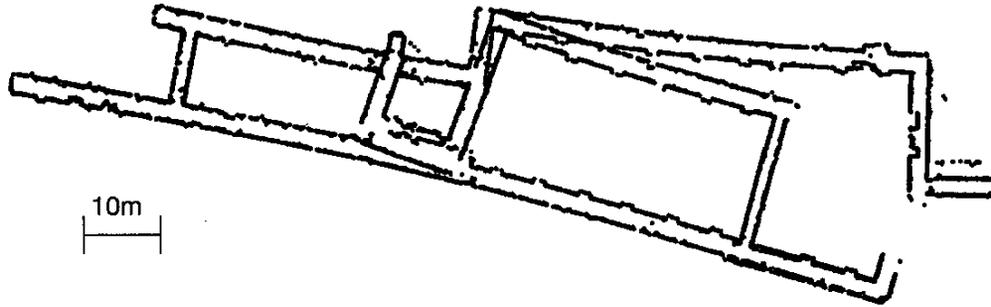


Fig. 7. The results of mapping a 5000 square meter environment This map is not adjusted for dead reckoning error.

The arc carving procedure can now be defined as follows: given a temporally ordered set of arcs $0 - n$, with regions of possible reflection $C_0 - C_n$ and freespace regions $F_0 - F_n$, the updated regions of possible reflection C'_i are defined as

$$C'_i = C_i - \bigcup_{j \geq i} Z_{ij} \quad (3.5)$$

C'_i is the new truncated arc, with angular endpoints θ'_1 and θ'_2 . If C'_i is connected over $[\theta'_1, \theta'_2]$, P'_i is defined as

$$P'_i = \begin{bmatrix} x + r \cos(\frac{\theta'_1 + \theta'_2}{2}) \\ y + r \sin(\frac{\theta'_1 + \theta'_2}{2}) \end{bmatrix} \quad (3.6)$$

It is possible that by repeated carving C'_i will become the empty set. This occurs if every point that C_i indicates as a possible point of reflection is contained inside the freespace region of a more recent arc. This indicates that either C_i corresponds to an incorrect reading, or that the obstacle that produced C_i is no longer present. In either case, it is no longer desirable or meaningful to add P_i to the local map. In this way, arc carving often completely eliminates spurious sonar returns.

It is also possible that carving may result in C'_i splitting into two or more disconnected regions. Over time, we have observed that this occurs when the F_j overlapping C_i corresponds to an arc j whose origin $(x, y)_j$ is closer to C_i than is $(x, y)_i$. Since the robot has moved closer to the obstacle corresponding to C_i , subsequent arcs C_k from the same obstacle are more accurate than C_i (Equation 5.2). Since readings exist that are more recent and more accurate than reading i , C_i should be ignored. Ignoring these arcs also ensures that C'_i is always connected over $[\theta_1, \theta_2]$. Thus, the carving process becomes simply reducing the range $[\theta_1, \theta_2]$ of C_i to avoid overlaps with freespace regions. This also ensures that P'_i can always be directly computed from the 5-tuple $(x, y, r, \theta_1, \theta_2)_i$

4. INTEGRATION OF ARC CARVING AND OTHER METHODS

Although an independent sonar processing technique, arc carving can be used in conjunction with previously discussed methods. Specifically, arc carving provides significant improvement to the average azimuth resolution when compared

to the centerline model. In the worst case, the two are equivalent. Arc carving does not significantly increase latency as compared to centerline.

In arc carving, every sonar reading begins using the same sonar model, an arc of width ϕ . As arcs are carved, each reading is updated to have its own model that is consistent with more recent readings. This idea can be extended to occupancy grids. Rather than use the same probability distribution for every sonar reading, a new distribution can be used for each reading, computed from the carved results of recent readings from a particular sensor. This would result in a more refined update of grid cells.

Arc carving can be used to enhance existing line fitting techniques. Increasing the accuracy of the location of points in the local map allows for fitting more accurate lines with less data. By eliminating spurious sonar readings, arc carving can provide local maps with less noise, thereby further facilitating line fitting. Similarly, reduction in noise can aid in more accurate clustering of points. Also, decreasing the azimuth uncertainty of a sonar arc results in fewer possible lines tangent to that arc, which can provide a significant enhancement to McKerrow's techniques.

Finally, arc carving can be efficiently run in parallel with ATM, as described in the next section. This allows for com-

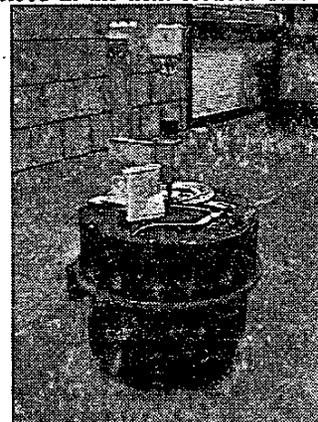


Fig. 8. Slammer, a Nomadic Scout2 with 16 equally spaced Polaroid ultrasonic sensors

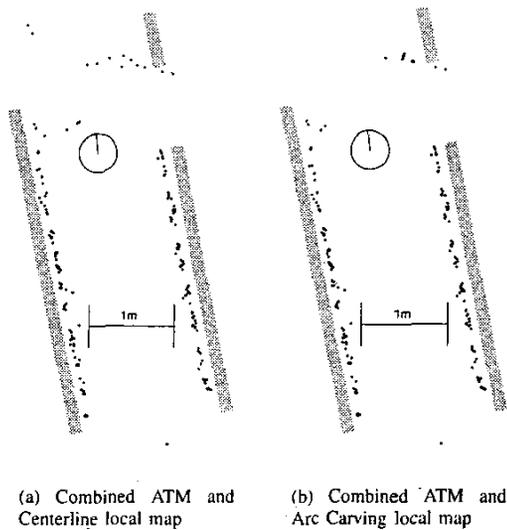


Fig. 9. Local maps produced as the robot navigates a narrow hallway. Light grey obstacles have been drawn for the sake of display. In the combined ATM and Centerline map, the hallway incorrectly appears to terminate.

binning the most accurate data available from each technique to produce an improved and timely local map. This local map will have low latency due to the availability of arc carved data, and high resolution due to the accuracy of both techniques. Comparisons between the resolution of ATM and arc carving are made in Section 5-C.

5. EXPERIMENTAL RESULTS

A. Implementation

The implementation of arc carving is straight forward. A finite history of arcs is kept, each stored as the 5-tuple $(x, y, r, \theta_1, \theta_2)$. These values are relative to the world coordinate frame, not relative to the robot since the robot will move between readings. Whenever a new arc C_i is added, each previous arc in the history is checked to see if it overlaps F_i . If

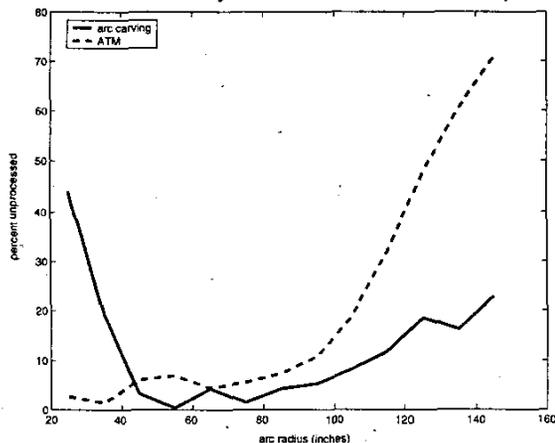


Fig. 10. The percentage of data that is unprocessed by the ATM and arc carving approaches, plotted by distance of the sonar return

so, then new θ_1 and θ_2 are calculated that remove this overlap. If a previous arc is completely contained within F_i , or if it would become disconnected due to carving, it is removed from the history. When an updated local map is requested one point is computed per arc in the history according to Equation 3.6. If an arc C_i is uncarved, then $P'_i = P_i$, and the point returned is labeled as a point generated by the centerline model. Once an arc is initialized, the only values that change are the values θ_1 and θ_2 . x, y and r remain unchanged. Since the beam width ϕ is constant for a given sensor, the only additional information that needs to be stored for ATM is the original sensor angle θ . By storing the 6-tuple $(x, y, r, \theta, \theta_1, \theta_2)$, arc carving and ATM can be run in parallel using the same data, without interfering with each other. The separate results of each technique can then be added to a combined local map.

Arc carving and ATM, along with our hierarchical SLAM approach [1], were implemented on a Nomadic Scout2 robot named Slammer (Fig. 8). Slammer has 16 Polaroid range sensors, equally spaced around its circumference. Local maps are used to trace out the generalized Voronoi graph[14], and to aid in topological localization, as described in [15]. Platform specific issues encountered during implementation are addressed in the appendix.

Sonar data was collected and processed in real time during SLAM trials in a building environment spanning more than 5000 square meters (Fig. 7). Qualitatively, the performance of arc carving can be described by the success or failure of the robot's navigation task. Fig. 9 shows one of many examples of arc carving having a positive effect on the robot's ability to navigate. In Fig. 9(a), the corridor appears too narrow to continue based only on ATM and Centerline data. Specifically, the points extending away from the left wall in front of the robot, and away from the top corner of the opening on the right, incorrectly make the hallway appear impassable. The points extending from the left wall correspond to arcs whose left endpoint is on the left wall. The points extending from the top corner correspond to arcs whose right endpoint is at the top corner. However, because the centerline model uses the center of the arc as the point of reflection, the corresponding points in the local map extend inside the hallway. Higher resolution ATM data is not yet available for that section of the map. However, based on ATM and arc carved data (Fig. 9(b)), the robot can continue to navigate the corridor. This is one of many examples of arc carving aiding the robot in navigation and coverage.

B. Latency

The sonar history is a ring buffer: arcs are being continuously added, each new arc replacing the oldest one. All arcs in the history are considered either processed or unprocessed with respect to each technique being employed. For example, with the centerline approach, all arcs are considered processed since no interactions between arcs are required. In the case of techniques that fuse multiple readings, an arc must first interact with other arcs in the manner prescribed by that technique before it can be considered processed. It is this

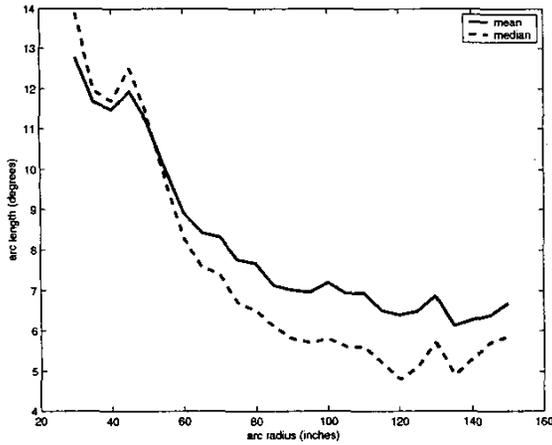


Fig. 11. The mean and median angular lengths of carved arcs

TABLE I

THE MEAN PERCENTAGE OF THE HISTORY THAT IS UNPROCESSED

Method	Percentage
ATM	31.75
Arc Carving	55.59
Arc Carving and ATM	7.68
Centerline	0

delay that results in increased latency. Therefore, latency for a particular technique can be measured by determining what percentage of arcs in the history are still unprocessed with respect to that technique. A higher percentage of unprocessed arcs corresponds to higher latency. These results are shown in Table I.

At first glance, these numbers would seem to indicate that arc carving has higher latency than ATM. However, this is not correct. In actuality, the average latency has been artificially weighted due to the high frequency of small radius arcs, which do have high arc carving latency. This is depicted in Fig. 10, which plots the experimentally recorded relationship between latency and the radius of a sonar return for both arc carving and ATM. As predicted by Equation 2.2, ATM latency increases proportional to the radius of the arc. However, arc carving latency remains low in almost all cases, the exception being the smallest radii. This is because small arcs have small path lengths, making it unlikely that a region of freespace from another arc will overlap. It is only this set of small radii arcs for which arc carving will have latency problems. Small radii arcs are inherently more accurate (Equation 5.2), making the centerline model sufficient in practice for such arcs. Conversely, large radii arcs are inherently inaccurate, and therefore it is important that they be processed. For these arcs, arc carving has far lower latency than ATM. In this way, arc carving and ATM complement each other nicely, resulting in on average more than 92 percent of all arcs used in the local map receiving at least some increase in azimuth resolution.

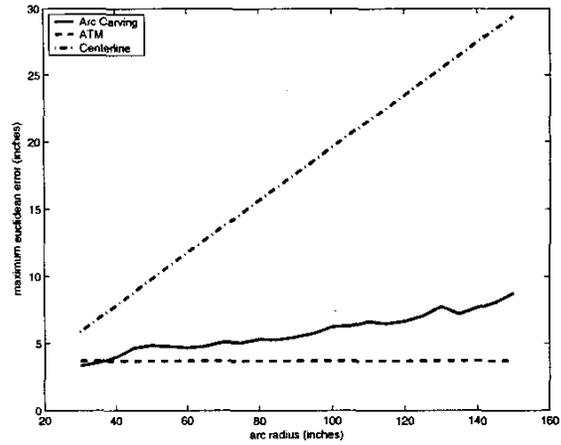


Fig. 12. The mean maximum euclidean error

C. Azimuth Resolution

The angular length of a carved arc is the size of the range $[\theta_1, \theta_2]$, denoted by

$$\theta_d = |\theta_2 - \theta_1| \quad (5.1)$$

Since P_i falls in the center of this range, the maximum possible angular error is

$$E_m = \frac{|\theta_2 - \theta_1|}{2} = \frac{\theta_d}{2} \quad (5.2)$$

Therefore, the azimuth accuracy is determined solely by the value of θ_d . As the radius of a sonar return increases, so does the path length of C_i . This results in increased likelihood of C_i being carved, and a decrease in θ_d (Fig. 11).

Given that P_i could be displaced around $(x, y)_i$ by as much as E_m , then the maximum euclidean error is the chord on a circle of radius r with central angle E_m .

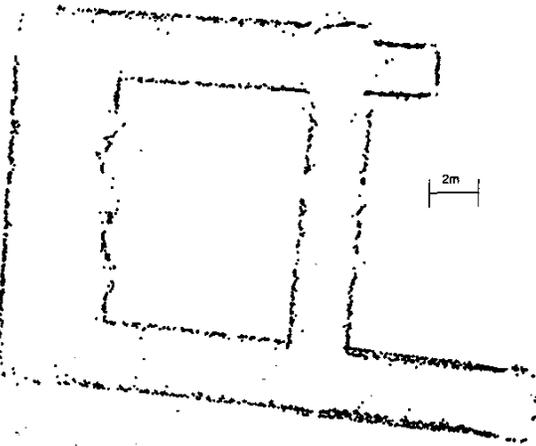
$$E_s = 2r \sin\left(\frac{E_m}{2}\right) \quad (5.3)$$

Fig. 12 shows E_s calculated from experimental data for arc carving, along with the theoretical values for ATM and centerline. E_s for centerline is calculated in the same manner as for arc carving, with E_m equal to the constant $\frac{\phi}{2}$, where ϕ is the beam width. The theoretical maximum error for ATM is

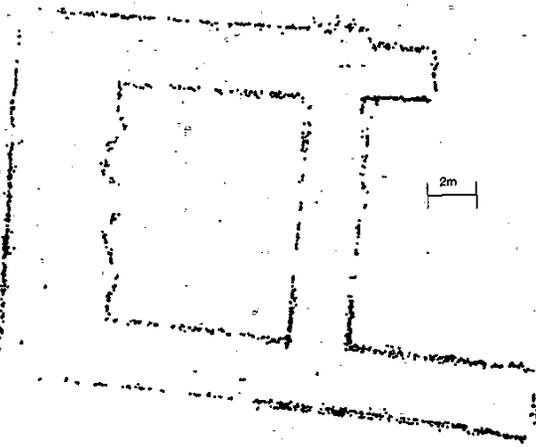
$$\frac{1}{\sin(\psi)} + \frac{1}{\tan(\psi)} \quad (5.4)$$

where ψ is the minimum transversal angle.

This data was collected using a history of 250 arcs. Increasing the length of the history can result in an even further improvement in resolution, since there is a longer window of opportunity for an arc to be carved. However, both arc carving and ATM are $O(mn)$ for m sonar readings and a history length of n , since every new arc must be checked against every previous arc for either intersections or overlap with regions of freespace. Therefore, any improvement in resolution due to a longer history must be balanced against the computational



(a) ATM map



(b) Arc Carving map (with maximum error thresholding)

Fig. 13. Final filtered maps produced by ATM and arc carving. These maps are not adjusted for dead reckoning error.

cost. Also, the longer the history, the longer points in the local map could remain for an obstacle that is no longer being detected. A longer history makes the local map less responsive to dynamic environments.

These error estimates do not include arcs that have been fully carved, that is where C_i^j is the empty set. In our experiments over 20 percent of all arcs processed by arc carving were fully carved, and therefore were removed from the local map. This demonstrates that arc carving is effective in removing spurious sonar returns, an important step towards more robust navigation.

Fig. 13 shows global range maps produced by arc carving and ATM for a small environment. In Fig. 13(a), points correspond to the final median of transversal intersections

along each arc. In Fig. 13(b), points correspond to the center of sufficiently carved regions of possible reflection. These maps are quite similar, demonstrating that the final accuracy of arc carving is equivalent to that of ATM. However, the arc carving results were obtained in a timely fashion, enabling the robot to make more informed decisions with respect to navigation, localization, and coverage.

6. CONCLUSIONS

In this paper we introduced arc carving, a new technique for processing ultrasonic range data. Arc carving derives from a sonar return information relating to both the presence and absence of obstacles. Multiple readings are combined to produce continually refined regions that likely caused sonar reflections. Each region contributes one point to a local range map used for mobile robot navigation.

This iterative refinement process has three key effects. First, every sonar reflection from an obstacle continues to improve the accuracy of the local map. Second, arc carving is often able to fully eliminate points corresponding to spurious sonar returns. Third, the robot never has to wait for the quality or quantity of sonar returns to pass a threshold before processed data is available. This immediate availability of accurate range data is essential for robust mobile robot navigation and coverage.

Arc carving can be used independently, or in conjunction with other processing techniques. The data structures involved in implementing arc carving are similar to those used by the ATM approach. This allows for a straightforward parallel implementation that lets each method's strengths complement the other's weaknesses. Using this combined implementation, our experimental platform has been able to navigate and map areas in excess of 5000 square meters.

In the future, we will explore the possibility of integrating ATM and arc carving further, allowing each method to make use of the intermediate results of the other. This would also allow selectively running only ATM or arc carving at one time, based on the particulars of the situation, resulting in a more computationally efficient approach. We will also explore the possibility of attaching certainty values to points in the local map. Both arc carving and ATM provide for an upper bound on the possible error of each point. Combining these with upper bounds on position stamping error (Appendix A) may allow for a meaningful estimate of the variance of sections or even the entire local map, which in turn could lead to even more robust navigation. Another issue to explore involves determining the optimal relationship between the length of the arc history, computational resources, and robot velocity. Finally, there is the issue of noise in the origin, orientation, and distance of a sonar return. Future work will investigate how to compensate for this noise.

ACKNOWLEDGMENTS

The authors would like to thank George Kantor for his assistance in this work.

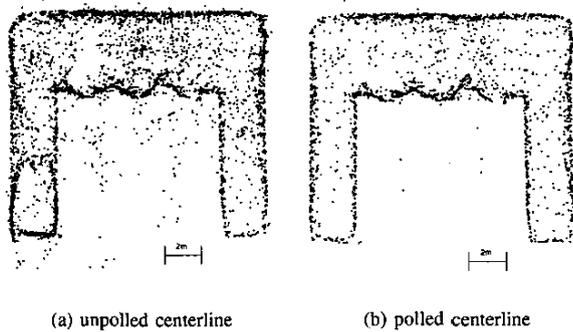


Fig. 14. Centerline generated maps from the same trial, with and without sonar polling. No additional filtering has been performed.

REFERENCES

- [1] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, vol. 1, Oct. 2003, pp. 448–453.
- [2] H. Choset, K. Nagatani, and N. Lazar, "The arc-transversal median algorithm: an approach to increasing ultrasonic sensor accuracy," 1999.
- [3] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," Computer Science Dept., U. Rochester, Tech. Rep. TR692, 1998.
- [4] S. Ciarcia, "An ultrasonic ranging system," *Byte Magazine*, pp. 113–123, October 1984.
- [5] J. Leonard and H. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Norwell, MA: Kluwer, 1992.
- [6] J. Budenske and M. Gini, "Why is it so difficult for a robot to pass through a doorway using ultrasonic sensors?" in *IEEE International Conference on Robotics and Automation*, 1994, pp. 3124–3129.
- [7] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3(3), pp. 249–265, June 1987.
- [8] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *IEEE Int. Conf. on Robotics and Automation*, 1985.
- [9] J. Borenstein and J. Koren, "Histogram in-motion planning for mobile robot obstacle avoidance," *IEEE Journal of Robotics and Automation*, vol. 7, pp. 535–539, 1991.
- [10] D. Pagac, E. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 623–629, August 1998.
- [11] P. McKerrow, "Echolocation—from range to outline segments," *Robot. Auton. Syst.*, vol. 11, no. 4, pp. 205–211, 1993.
- [12] J. Leonard, B. Moran, I. Cox, and M. Miller, "Underwater sonar data fusion using an efficient multiple hypothesis algorithm," in *Proc. of IEEE Conference on Robotics and Automation*. Nagoya, Japan: IEEE Press, May 1995, pp. pages 2995–3002.
- [13] P. MacKenzie and G. Dudek, "Precise positioning using model-based maps," in *Proceedings of the International Conference on Robotics and Automation*. San Diego, CA: IEEE Press, 1994.
- [14] H. Choset and J. Burdick, "Sensor based planning. part 2: Incremental construction of the generalized voronoi graph," in *Proc. of IEEE Conference on Robotics and Automation*. Nagoya, Japan: IEEE Press, May 1995, pp. 1643 – 1648.
- [15] K. Nagatani, H. Choset, and S. Thrun, "Towards exact localization without explicit localization with the generalized Voronoi graph," in *IEEE Int. Conf. on Robotics and Automation*, Lueven, Belgium, May 1998, pp. 342–348.

APPENDIX

A. Position Stamping without Hardware Support

The discussion up to now has assumed that every sonar return is accurately position stamped, that is the origin (x, y) and sensor orientation θ for each sonar return relative to the world coordinate frame (discounting the effects of positioning error)

are well known. When a robot's hardware provides support for position stamping, this assumption is valid. However, when hardware support is not provided, it is exceedingly difficult to know exactly when a sonar was fired, and therefore where the origin of that sonar arc is. This is the case with the Nomadics Scout2 robot. Since any error in position stamping is passed on to the local map, it is important to minimize this error, especially in orientation.

Our solution to this problem was a technique we call Sonar Polling. Given sensors labeled 1 to n in the order in which they fire, a record of each sensor's last reading is maintained, as well as which sensor, labeled l , is the last one known for sure to have fired. The sensors are then constantly polled to detect any changes in value from their last recorded value. If a sensor i changes value, then not only do we know that it has fired, but that all sensors between l and i must also have fired, even if they have not changed value. Also, based on how much time has passed since l was fired, it can be determined which sensors subsequent to l must have fired due to their timeout. In this way l is continuously updated, and the sonar returns can be more accurately position stamped. This method also results in a known upper bound on the position stamping error. When this bound grows too large, the robot can either choose to ignore readings with high uncertainty, or slow down to reduce the error. Fig. 14 shows the effect sonar polling alone can have on centerline processed sonar data.

B. Filtering

Another problem with the Scout2 hardware are spurious sonar readings. Such readings can result by detecting an echo from a previously fired sensor. Another cause is the sonar beam bouncing off the floor. In the case of the Scout2, with the sensors mounted at only 10.25 inches off of the ground, a sonar beam will begin to reflect from the floor at just over 2 feet from the robot. When the robot tilts even slightly due to bumps or depressions in the floor, this problem worsens. Although arc carving can remove some of this noise by completely carving arcs, it is not always enough.

Our solution was to introduce two levels of filtering above and beyond any explicit sonar processing. The first was on a per-sensor basis: if the range readings coming from any one sensor varied by more than a threshold between cycles, that sensor would be ignored for one cycle. A maximum range of 150 inches was also used. Sonar returns beyond this distance were ignored. The second level of filtering is performed on the local map. All points must pass an outlier test based on there existing enough neighboring points within a specified radius. Isolated points are ignored until they either expire from the history, or more points are added around them.