Exploring an Unknown Polygonal Environment with Bounded Visibility

Amitava Bhattacharya¹, Subir Kumar Ghosh¹, and Sudeep Sarkar²

¹ School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India

email: {amitava@tcs.tifr.res.in, ghosh@tifr.res.in}

² *** Department of Computer Science and Engineering, University of South Florida, Tampa, USA 33620
email: {sarkar@csee.usf.edu}

Abstract

This paper integrates constraints from visual processing and robot navigation into the well-studied computational geometry problem of exploring unknown polygonal environments with obstacles. In particular, we impose two constraints. First, we consider a robot with limited visibility, which can reliably “see” its environment only within a certain range R . Second, we allow the computation of visibility only from discrete number of points on the path. Both these constraints arise from real life cost constraints associated with robotic exploration and robot vision processing. We present an online algorithm for exploration under such constraints and show that the maximum number of views needed to explore the unknown polygon (with obstacles) P , is bounded by $\left\lfloor \frac{2 \times \text{Perimeter}(P)}{R} \right\rfloor + \left\lfloor \frac{2 \times \text{Area}(P)}{R^2} \right\rfloor + 3n$, where n is the number of sides of the polygon. We also show that the competitive ratio of the algorithm is $6\pi + \frac{4r+3n}{\max(\left\lceil \frac{\text{Area}(P)}{\pi R^2} \right\rceil, \left\lceil \frac{\text{Perimeter}(P) - 2Rr}{2\pi R} \right\rceil)}$.

Keywords: Bounded visibility, Discrete visibility, Polygonal environment, Exploration, On-line algorithm

1 Introduction

The context of this work is the exploration of *unknown* polygonal environments *with* obstacles. Both the outer boundary and the inside obstacles boundaries are piecewise linear. The boundaries can be non-convex. Can we construct a complete representation of the environment, in terms of say its triangulation, as we explore it incrementally starting from a given position? Most online computational geometry algorithms for exploring unknown polygons [1,2,4,5,6,7,14,15,19,20,25], assume that the visibility region can be determined in a continuous fashion from each point on a path and that we have infinite visibility. While such

*** This work was done when the author visited Tata Institute of Fundamental Research on sabbatical leave from the University of South Florida, Tampa.

assumptions are reasonable in the case of a human “watchman,” they are not practical in the context of real-life robot navigation. First, autonomous robots can only carry a limited amount of on-board computing capability. At the current state of the art, computer vision algorithms for computing visibility polygons are time consuming [3,9,16,21]. The computing limitations dictate that it is not practically feasible to continuously compute visibility polygons along the robot’s trajectory. Second, for good visibility, the robot’s camera will be typically mounted on a mast. Such devices vibrate during movement, and hence for good precision (which is required to compute an accurate visibility polygon) the camera must be stationary for each view. Third, computer vision range sensors or algorithms, such as stereo or structured light range finder, can reliably compute the 3D scene locations only up to a depth R . The reliability of depth estimates is inversely related to the distance from the camera. Thus, the range measurements from a vision sensor for objects that are far away are not at all reliable. This suggests that it is necessary to modify exploration algorithms to make them more realistic by restricting visibility polygons by the range distance R . Therefore, the portion of the boundary of a polygonal environment within the range distance R is only considered to be visible from the camera of the robot. We refer to the visibility polygon under this range restriction as the *restricted visibility polygon*. Observe that restricted visibility polygons need not be always a closed boundary. In fact, it can consist of several disjoint polygons chains. So, exploring an unknown polygonal environment using restricted visibility polygons requires more number of views. In earlier works, Ghosh and Burdick [13,11,12] have presented algorithms that take into account the first two constraints. In this paper, we generalize their algorithm to take into account the constraint of bounded visibility and analyze its computational consequences.

The essential components that contribute to the total cost, in terms of time, required for a robotic exploration can be analyzed as follows [13]. Each move will have two associated costs. First, there is the time required to physically execute the move. If we crudely assume that the robot moves at a constant rate, r , during a move, the total time required for motion will be $\frac{d}{r}$, where d is the total path length followed by the robot during the exploration. Second, there are the exploratory processes in which the robot plans its moves based on its most recent geometric information about the scene. This cost has two subcomponents: time spent by the on board sensors to acquire information about the scene and the time spent to plan its next move. Let the average times spent on acquiring sensory information and planning be t_S and t_M , respectively, per operation. Let N_M and N_S be respectively the number of moves and the number of sensor operations that are required to complete the exploration of P . Then the total time, T , required to explore is

$$T(P) = t_M N_M + t_S N_S + \frac{d}{r} \quad (1)$$

In practice, the first two terms ($t_M N_M + t_S N_S$) is indeed a significant fraction of $T(P)$. Thus, we would like any exploration algorithm to minimize N_M and N_S . In this paper, we present an on-line algorithm that requires at most

$\left\lfloor \frac{2 \times \text{Perimeter}(P)}{R} \right\rfloor + \left\lfloor \frac{2 \times \text{Area}(P)}{R^2} \right\rfloor + 3n$ views for exploring an unknown polygonal environment.

In the next section, we review the relevant prior work. In Section 3 we present the Ghosh and Burdick algorithm, which needs only discrete set of visibility computations but does not handle bounded visibility. In Section 4 we present a generalized version of the exploration algorithm for bounded visibility in and prove its correctness. We also establish an upper bound on the number of views required to explore the entire region. In Section 5 we conclude the paper with a few remarks.

2 Prior Work

There have been many algorithms proposed for robot navigation in unknown environments. Rao [23] considered problem of navigating *generalized* polygons that have both straight and curved edges. Taylor and Kriegman [26] present landmark based exploration algorithms that are heavily dependent on vision sensors. Range sensor based navigation algorithm include that proposed by Foux *et al.* [10], Kamon *et al.* [17,16], and Ekman *et al.* [8]. The exploration strategy of Ekman *et al.* [8] is closest to that proven to be optimal by Ghosh and Burdick [13, 11,12]. Ekman *et al.* assumes a point robot with an ideal range sensor that can see up to infinity and can measure range in N uniformly distributed directions. Thus, the range sensors offer a sampled version of the visibility polygon. The edges of the visibility polygon that do not correspond to environmental edges are termed as the jump edges, which are used to suggest the next exploration point. Ghosh and Burdick proves that such a strategy is indeed an optimal one and is guaranteed to completely explore the polygonal environment.

The consideration of bounded visibility in computational geometry algorithms is rare. We could locate just one work that also uses the notion of bounded visibility but in a different context. Kim *et al.* [18] coined the notion of d-visibility to describe the situation when a robot as restricted range of sight. They presented optimal algorithms to find the edge visibility polygon under the d-visibility constraint.

3 Ghosh-Burdick Algorithm

In this section, we summarize the exploration algorithm of Ghosh and Burdick for a point robot. The robot starts at any initial location, p_1 , where it determines the visibility polygon $\mathcal{V}(P, p_1)$. Using the visible vertices of P in $\mathcal{V}(P, p_1)$, the robot triangulates as much of the $\mathcal{V}(P, p_1)$ as possible. Let this triangulation be denoted $T(P)$. The robot then executes a forward move to another position $p_2 \in \mathcal{V}(P, p_1)$ and computes the next visibility polygon $\mathcal{V}(P, p_2)$. The region common to $\mathcal{V}(P, p_2)$ and $T(P)$ is removed. The remaining free space in $\mathcal{V}(P, p_2)$ is triangulated and added to $T(P)$. This process is repeated till the entire free space is explored. It may appear that to see the entire free space it is enough

to see all the vertices and edges of P . But it is not the case as shown by the example in Figure 1.

Fig. 1. Three views are enough to see all vertices and edges but not the entire free space.

Fig. 2. The spiral polygon can be explored in $r+1$ steps.

In the following, we present the major steps of their algorithm.

Step 1: $i := 1$; $T(P) := \emptyset$; $S := \emptyset$; Let p_1 denote the starting position of the robot.

Step 2: Compute $\mathcal{V}(P, p_i)$; $T(P) := T(P) \cup T'(P)$ where $T'(P)$ is the triangulation of $\mathcal{V}(P, p_i)$; $S = S \cup p_i$;

Step 3: While $\mathcal{V}(P, p_i) - T(P) = \emptyset$ and $i \neq 0$ then $i := i - 1$;

Step 4: If $i = 0$ then goto Step 7;

Step 5: If $\mathcal{V}(P, p_i) - T(P) \neq \emptyset$ then choose a point z on any constructed of $\mathcal{V}(P, p_i)$ lying outside $T(P)$;

Step 6: $i := i + 1$; $p_i := z$; goto Step 2;

Step 7: Stop.

Ghosh and Burdick have shown that there exists at least a triangle every time $T'(P) - T(P)$ is computed and the algorithm computes at most $r+1$ views, where r is the total number of reflex vertices in P . It can be seen that $r+1$ views are also necessary as in Figures 2 and 3. We have the following theorem.

Theorem 1. *The algorithm of Ghosh-Burdick can be used to explore an unknown polygonal environment using restricted visibility polygons if $R \geq D$ where D is the longest line segment which lies completely inside the polygon.*

Proof: The proof follows from the fact that if $R \geq D$, any visibility polygon computed by the algorithm of Ghosh-Burdick is same as the restricted visibility polygon. $\square$

Fig. 3. The polygon is explored in $r+1$ steps.

Fig. 4. $RVP(p_i) = (ABCDEFGHA)$.

4 Generalization of Ghosh-Burdick Algorithm

In this section we present our exploration algorithm and establish an upper bound for its competitive ratio. In Theorem 2.1 we have shown that Ghosh-Burdick algorithm can be used for exploring an unknown polygonal environment when $R \geq D$. We now consider the case when $R < D$. Let RVP denote the restricted visibility polygon computed from the same point.

Fig. 5. A sequence of restricted visibility polygons.

Fig. 6. A line segment of length can cut at most $\left\lfloor \frac{\sqrt{2}L}{l} \right\rfloor + 3$ squares.

Observe that a restricted visibility polygon may not be closed (see Figure 4) and therefore its boundary can have circular edges in addition to constructed and polygonal edges. So it is necessary to take another view from some point in

RVP in order to see more of P . The process can be repeated till the union of these restricted visibility polygons cover P . Let the restricted visibility polygon computed from the point p_i be denoted by $RVP(p_i)$. The boundary of any region B is denoted by $bd(B)$. In the following we present a procedure to explore the polygon starting from a point p_0 using restricted visibility polygons (see Figure 5). The robot initializes its polar co-ordinate system by setting its origin at p_0 . Let $CP(p_0)$ denote the region of P so far visible from the robot.

Algorithm $\mathcal{A}$:

Step 1: $CP(p_0) := RVP(p_0)$; if the boundary of $CP(p_0)$ consists of only polygonal edges then goto Step 4.

Step 2: Choose a point p on any circular arc or constructed edges of $CP(p_0)$ and compute $RVP(p)$; $CP(p_0) := CP(p_0) \cup RVP(p)$.

Step 3: If $CP(p_0)$ has circular arc or constructed edges, then goto Step 2.

Step 4: Stop

Let us now prove the correctness of the algorithm. In order to prove its correctness we have to show $CP(p_0) = P$ when the algorithm terminates. It is obvious that $CP(p_0) \subset P$. So to show equality it is enough to show $P \subset CP(p_0)$. Assume on the contrary that $P \not\subset CP(p_0)$. Then there exists a point $p \in P$ and $p \notin CP(p_0)$. If there exists a path from p to p_0 lying inside $CP(p_0)$, then p belongs to $CP(p_0)$, a contradiction. Otherwise any path from p to p_0 must intersect $bd(CP(p_0))$. Since every edge of $bd(CP(p_0))$ is a polygonal edge, every path from p to p_0 must intersect $bd(P)$ a contradiction. Hence $CP(p_0) = P$ when the algorithm terminates.

In the following lemmas we establish an upper bound for the number of views required by a robot to explore the region P using $\mathcal{A}$.

Lemma 2. *If a line segment of length L lies on a grid of size l , then the number of squares that can be cut on one side is at most $\left\lfloor \frac{\sqrt{2}L}{l} \right\rfloor + 3$ (see Figure 6).*

Proof : Let θ be the slope of the line. Then the number of vertical lines it can cut is at most $\left\lfloor \frac{L \cos \theta}{l} \right\rfloor + 1$. Similarly the number of horizontal lines it can cut at most is $\left\lfloor \frac{L \sin \theta}{l} \right\rfloor + 1$. Observe that each time it cuts a vertical line or a horizontal line it enters a new square. Therefore, the maximum number of squares it can cut is $\left\lfloor \frac{L(\sin \theta + \cos \theta)}{l} \right\rfloor + 3$. Note that 1 is added to account for the starting cell. Since the maximum value of $\sin \theta + \cos \theta = \sqrt{2}$, the maximum number of squares it can cut on one side is $\left\lfloor \frac{\sqrt{2}L}{l} \right\rfloor + 3$. $\square$

Lemma 3. *If the area, perimeter and number of vertices of a polygon P are $\text{area}(P)$, $\text{perimeter}(P)$ and n respectively, then the number of views required using algorithm $\mathcal{A}$ is bounded by $\left\lfloor \frac{2 \times \text{Perimeter}(P)}{R} \right\rfloor + \left\lfloor \frac{2 \times \text{Area}(P)}{R^2} \right\rfloor + 3n$.*

Proof : Suppose there exists a partition of the polygon into cells such that inside each cell the robot can take at most 1 view. Then the number of cells in any such partitions will give an upper bound on the number of views required. Using this approach an upper bound is obtained as follows. The polygon P is

Fig. 7. A polygon P placed on a grid.

Fig. 8. The polygon can be explored in $r+1$ views.

placed on a grid where each cell is a square of size $\frac{R}{\sqrt{2}}$ (see Figure 7). Consider the squares that lie totally inside P . Clearly the number of such squares is at most $\frac{2 \times \text{Area}(P)}{R^2}$. Observe that the robot can take at most one view in each such square since the size of the grid is $\frac{R}{\sqrt{2}}$. So the robot can take at most $\frac{2 \times \text{Area}(P)}{R^2}$ views from such squares. Let us count the number of views that can be taken from squares which intersect the polygon boundary. Since the grid size is $\frac{R}{\sqrt{2}}$, by Lemma 4.1, a side of length L can cut at most $\lfloor \frac{2L}{R} + 3 \rfloor$ squares. Consider the squares which are cut by the boundary. The part of these squares which lie inside the polygon can be covered by convex regions with the straight line parts of the polygon boundary as one of its sides. For any such convex region, at most one view can be taken, since the diameter of such regions is less than $\frac{R}{\sqrt{2}}$. So the maximum number of views the robot can take is $\left\lfloor \frac{2 \times \text{Perimeter}(P)}{R} \right\rfloor + \left\lfloor \frac{2 \times \text{Area}(P)}{R^2} \right\rfloor + 3n$. $\square$

Now we derive the competitive ratio for algorithm $\mathcal{A}$. Observe that in each view a robot can see at most πR^2 area of P and $2\pi R + 2Rr'$ of the boundary of P , where r' is the number of reflex vertices seen in that view. Hence even in the best case the robot must take at least $\max(\left\lceil \frac{\text{Area}(P)}{\pi R^2} \right\rceil, \left\lceil \frac{\text{Perimeter}(P) - 2Rr}{2\pi R} \right\rceil)$ views. Hence the competitive ratio is

$$\frac{\left\lfloor \frac{2 \times \text{Perimeter}(P)}{R} \right\rfloor + \left\lfloor \frac{2 \times \text{Area}(P)}{R^2} \right\rfloor + 3n}{\max(\left\lceil \frac{\text{Area}(P)}{\pi R^2} \right\rceil, \left\lceil \frac{\text{Perimeter}(P) - 2Rr}{2\pi R} \right\rceil)}$$

which is upper bounded by $6\pi + \frac{4r+3n}{\max(\left\lceil \frac{\text{Area}(P)}{\pi R^2} \right\rceil, \left\lceil \frac{\text{Perimeter}(P) - 2Rr}{2\pi R} \right\rceil)}$.

The worst case arises when the number of reflex vertices is large. Then the number of views required is $r+1$ but two views are sufficient as shown in Figure 8.

5 Concluding Remarks

Observe that when a view is taken from some cell, some portions of other cells may be inside the restricted visibility polygon. Since this fact is not incorporated in calculating the bound, we expect that the number of views will be much less in practice. We also note that as R increases, we need fewer views which is reflected in the bound.

The exploration problem may be considered as a covering P with circles [27] with the additional constraint that between centers a and b of any two circles there is a path composed of line segments $au_1, u_1u_2, \dots, u_ib$, such that 1) each of the points $u_1, u_2, \dots, u_i$ are centers of circles, 2) each segment lies completely in the polygon P and 3) length of each segment is at most R .

Suppose $p_1, p_2, \dots, p_k$ be the points of P such that an optimal exploration algorithm for a point robot has computed restricted visibility polygons from these points. We know that $\bigcup_{i=1}^k RVP(P, p_i) = P$, $p_{i+1} \in V(P, p_i)$ and k is minimum. So, P can be guarded by placing stationary guards at $p_1, p_2, \dots, p_k$ where it is assumed that each guard can see up to distance R . Hence the exploration problem for a point robot, with limited visibility is the Art Gallery problem with stationary guards [22, 24] with the additional constraints. Thus, our exploration algorithm for a point robot is an approximation algorithm for this variation of the Art Gallery problem which also seems to be NP-hard.

References

1. E. Bar-Eli, P. Berman, A. Fiat, and P. Yan. On-line navigation in a room. In *Proceedings of the third ACM-SIAM Symposium on Discrete Algorithms*, pages 237–249, 1992.
2. A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, pages 494–504, 1991.
3. J. Borenstein, H. R. Everett, and L. Feng. *Navigating mobile robots: sensors and techniques*. A. K. Peters Ltd., Wellesley, MA, 1995.
4. K. Chan and T. W. Lam. An on-line algorithm for navigating in an unknown environment. *International Journal of Computational Geometry and Applications*, 3:227–244, 1993.
5. A. Datta and C. Icking. Competitive searching in a generalized street. In *Proceedings of ACM Symposium on Computational Geometry*, pages 175–182, 1994.
6. X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment i: The rectilinear case. In *Proceedings of the 32nd IEEE Symposium on Foundation of Computer Science*, pages 298–303, 1991.
7. G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. In *Proceedings of the Sixth ACM-SIAM Symposium on Discrete Algorithms*, pages 437–446, 1995.
8. A. Ekman, A. Trone, and D. Stromberg. Exploration of polygonal environments using range data. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 27(2):250–255, April 1997.
9. O. Faugeras. *Three-dimensional computer vision*. MIT Press, Cambridge, 1993.

10. G. Foux, M. Heymann, and A. Bruckstein. 2-dimensional robot navigation among unknown stationary polygonal obstacles. *IEEE Transactions on Robotics and Automation*, 9(1):96–102, February 1993.
11. S. K. Ghosh and J. W. Burdick. An on-line algorithm for exploring an unknown polygonal environment by a point robot. In *Proceedings of the Ninth Canadian Conference on Computational Geometry*, pages 100–105, 1997.
12. S. K. Ghosh and J. W. Burdick. Understanding discrete visibility and related approximation algorithms. In *Proceedings of the Ninth Canadian Conference on Computational Geometry*, pages 106–111, 1997.
13. S. K. Ghosh and J. W. Burdick. Exploring an unknown polygonal environment with a sensor based strategy. *Submitted for publication*, 2000.
14. S.K. Ghosh and S. Saluja. Optimal on-line algorithms for walking with minimum number of turns in unknown streets. *Computational Geometry: Theory and Applications*, 8:241–266, 1997.
15. C. Icking and R. Klein. Searching for the kernel of a polygon—a competitive strategy. In *Proc. 11th Annu. ACM Symposium Comput. Geom.*, pages 258–266, Vancouver, Canada, June 1995.
16. I. Kamon, E. Rimon, and E. Rivlin. Tangentbug: A range sensor-based navigation algorithm. *International Journal of Robotics Research*, 17(9):934–953, September 1998.
17. I. Kamon and E. Rivlin. Sensory-based motion planning with global proofs. *IEEE Transactions on Robotics and Automation*, 13(6):814–822, December 1997.
18. S. H. Kim, J. H. Park, S. H. Choi, S. Y. Shin, and K. Y. Chwa. An optimal algorithm for finding the edge visibility polygon under limited visibility. *Information Processing Letters*, 53(6):359–365, March 1995.
19. R. Klein. Walking an unknown street with bounded detour. *Computational Geometry: Theory and Applications*, 1:325–351, 1992.
20. J. Kleinberg. On line search in a simple polygon. In *Proceedings of the fifth ACM-SIAM Symposium on Discrete Algorithms*, pages 8–15, 1994.
21. J. Leonard and H. F. Durrant-Whyte. *Directed sonar sensing for mobile robot navigation*. Kulwer Academic Publishers, Boston, MA, 1992.
22. J. O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, NY, 1987.
23. N.S.V. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):947–962, 1995.
24. T. Shermer. Recent results in art galleries. *Proc. of the IEEE*, 80(9):1384–1399, Sept. 1992.
25. I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21:863–888, 1992.
26. C. J. Taylor and D. J. Kriegman. Vision-based motion planning and exploration algorithms for mobile robots. —*IEEE Transactions on Robotics and Automation*, 14(3):417–426, June 1998.
27. G. Fejes Tóth. Packing and covering. In *Handbook of Discrete and Computational Geometry*, (edited by J. E. Goodman and J. O’Rourke), pages 19–42. CRC Press, 1997.