Online Algorithms for Searching and Exploration in the Plane

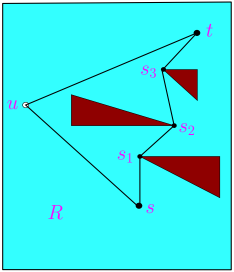
Subir Kumar Ghosh

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

Overview

1. What is online algorithm?
2. Efficiency of online algorithms.
3. Searching for a target on a line.
4. Searching for a target in an unknown region.
5. Continuous and discrete visibility.
6. Searching for a target in an unknown street.
7. Searching for a target in an unknown star-shaped polygon.
8. Exploring an unknown polygon: Continuous visibility.
9. Exploring an unknown polygon: Discrete visibility.
10. Exploring an unknown polygon: Bounded visibility.
11. Mapping polygons using mobile agents.

What is offline algorithm?



- Starting from s, a point robot is searching for the point t in R.
- If the robot has the complete geometric information (or map) of R and also knows the exact location of t, then the robot can choose a path inside R to move from s to t.
- In many situations, it is expected that the robot follows the Euclidean shortest path from s to t inside R.
- In some situation, the robot may be asked to follow a minimum link (or, turn) path from s to t inside R.

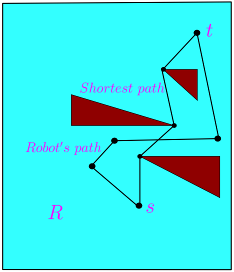
- There are known efficient sequential algorithms for computing such paths.
- Thus, the robot can compute an optimal path, depending upon the optimization criteria, using its on-board computer system and then follows the path from s to t.
- Such algorithms are called *offline algorithms* of a robot path planning for a target searching problem in a known environment.

1. S. K. Ghosh, *Visibility Algorithms in the Plane*, Cambridge University Press, United Kingdom, 2007.
2. J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.

What is online algorithm?

- Suppose, a robot does not have the complete knowledge of the geometry of R apriori.
- The robot also does not know the location of the target t, but the target can be recognized by the robot.
- In such a situation, the robot is asked to reach t from its starting position s using its sensory input provided by acoustic, visual, or tactile sensors of its on-board sensor system.
- The problem here is to design an efficient *online algorithm* which a robot can use to search for the target t.
- Observe that any such algorithm is 'online' in the sense that decisions must be made based only on what the robot has received input so far from its sensor system.

Efficiency of online algorithms



One of the difficulties in working with incomplete information is that the path cannot be pre-planned and therefore, its global optimality can hardly be achieved.

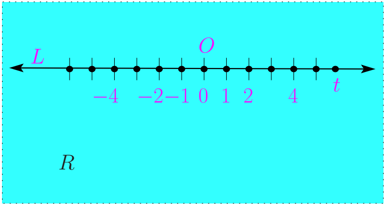
Instead, one can judge the online algorithm performance based on how it stands with respect to other existing or theoretically feasible algorithms.

The efficiency of online algorithms for searching and exploration algorithms is generally measured using their competitive ratios.

$$Competitive\ ratio = \frac{Cost\ of\ the\ online\ algorithm}{Cost\ of\ an\ optimal\ offline\ algorithm}$$

1. S. K. Ghosh and R. Klein, *Online algorithms for searching and exploration in the plane*, Computer Science Review, 4:189-201, 2010.
2. P. Berman, *On-line searching and navigation*, Lecture Notes in Computer Science 1442, pp. 232-241, Springer, 1996.
3. D. D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Communication of ACM, 28: 202-208, 1985.

Searching for a target on a line

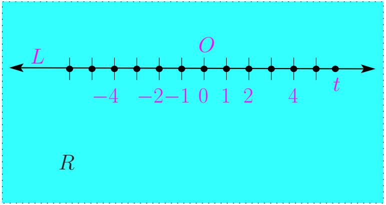


- Suppose, the target point t is placed on a line L in an unknown location.
- Starting from a given position O on L, the problem is to design an online algorithm for a point robot for locating t.
- It is assumed that the robot can detect t if it stands on top of t or reaches t.
- The problem may be viewed as an autonomous robot is facing a very long wall and it wants go to the other side of the wall through a door on the wall but it does not known whether the door is located to the left or right of its current position.

- Suppose the robot knows that t is located exactly d distance away from O.
- Then the robot first walks d distance to the right.
- If t is not found, then the robot returns to O and then walks d distance to the left.
- So, the competitive ratio of this straightforward on-line algorithm is 3.

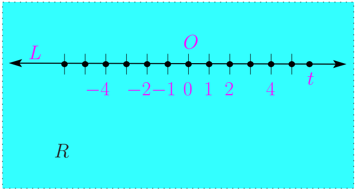
What is the competitive ratio of the search if d is not known apriori?

Alternate walk



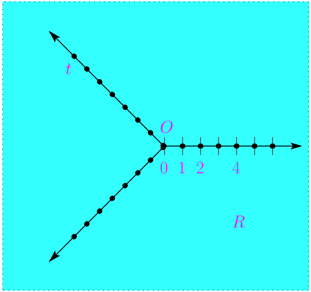
- The robot walks one unit to the right along L. If t is not found, then it returns to its starting point O.
- In the next step, the robot walks two units to the left of O along L. If t is not found again, the robot returns to O.
- In the next step, the robot walks four units to the right along L and if it is again unsuccessful to locate t, it returns to O.
- After some steps, the robot locates t.

The process of doubling the length is known as *doubling strategy*.



- Assume that t is located at a distance d from the origin on the positive axis.
- Assume that $2^{k-1} < d \leq 2^{k+1}$ for some k.
- The total distance traveled during the alternative walk is $(2.1 + 2.|-2| + 2.4 + 2.|-8| + \ldots + 2.2^{k-1} + 2.|-2^k| + d$ $= 2.2^{k+1} + d)$.
- If the location of t is known apriori, then it is a straight walk of length d from the origin to t.
- So, the competitive ratio of the alternate walk is $(2.2^{k+1} + d)/d = 1 + 2.2^{k+1}/d$ which is at most $1 + (2.2^{k+1}/2^{k-1}) = 9$.

Searching for a target on m rays



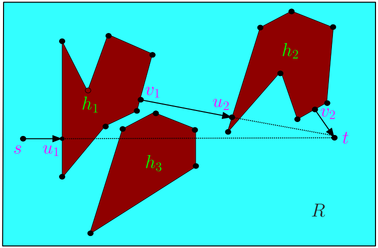
A beautiful young cow Ariadne is at the entrance of a simple labyrinth which branches in $m \geq 2$ corridors. She knows that the handsome Minotaur is waiting somewhere in the labyrinth. What is the best searching strategy for Ariadne to locate Minotaur?

1. S. Gal, *Minimax solutions for linear search problems*, SIAM Journal on Applied Mathematics, 27:17-30, 1974.
2. S. Gal, *Search games*, Academic Press, New York, 1980.

- Visit $m \geq 2$ rays in a cyclic order starting with an initial walk of length one.
- Increase the length of the walk each time by a factor of $m/(m-1)$ till t is located.
- This strategy gives the competitive ratio of $1 + 2m^m/(m-1)^{m-1}$, which is optimal.

1. R. A. Baeza-Yates, J. C. Culberson and G. J. E. Rawlins, *Searching in the plane*, Information and Computation, 106:234-252, 1993.
2. A. Eubeler, R. Fleischer, T. Kamphans, R. Klein, E. Langetepe and G. Trippen, *Competitive online searching for a ray in the plane*, Robot Navigation, Schloss Dagstuhl, Germany, 2006.
3. E. Langetepe, *On the optimality of spiral search*, Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, 2010.

Searching for a target in an unknown region



- Assume that the point robot knows the exact location of t but does not know the positions of unknown polygonal obstacles $h_1, h_2, \ldots, h_k$.
- The robot starts from s, and moves towards t following the segment st till the robot detects by its tactile sensor that it has hit a polygonal obstacle (say, h_i) at a some point u_i.
- Then the robot goes around the boundary of h_i to locate the boundary point of h_i (say, v_i) which is closest to t.
- Then the robots moves from u_i to v_i following the shorter of the two paths from u_i to v_i along the boundary of h_i .

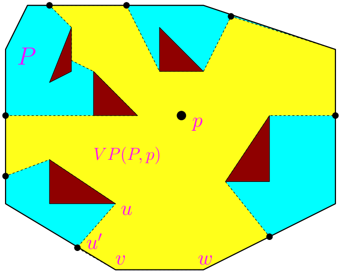
- Then the robots moves from u_i to v_i following the shorter of the two paths from u_i to v_i along the boundary of h_i .
- Treating v_i as s, the robot repeats the same process of moving towards t following the segment $v_i t$ till t is reached.
- The length of the path traversed by the robot is bounded by the length of st and 1.5 times the perimeters of those polygonal obstacles that are hit by the robot.

1. V. Lumelsky and A. Stepanov, *Dynamic path planning for a mobile automaton with limited information on the environment*, IEEE Transactions on Automatic Control, AC-31:1058-1063, 1986.
2. V. Lumelsky and A. Stepanov, *Path planning strategies for point automation moving amidst unknown obstacles of arbitrary shape*, Algorithmica, 2:402-430, 1987.

Algorithms for target searching in an unknown unbounded region

1. C. Papadimitriou and M. Yannakakis, *Shortest paths without map*, Theoretical Computer Science, 84:127-150, 1991.
2. A. Blum and P. Raghavan and B. Schieber, *Navigating in unfamiliar geometric terrain*, SIAM Journal on Computing, 26 (1997), 110-137.
3. P. Berman, A. Blum, A. Fiat, H. J. Karloff, A. Rosn and M. E. Saks, *Randomized robot navigation algorithms*, Proc. of the 7th ACM-SIAM Symposium on Discrete Algorithms, pp. 75-84, 1996.
4. E. Bar-Eli, P. Berman, A. Fiat and P. Yan, *On-line navigation in a room*, Journal of Algorithms, 17:319-341, 1994.
5. A. Mei and Y. Igarashi, *An efficient strategy for robot navigation in unknown environement*, Information Processing Letters, 52:127-150, 1994.

Visibility polygon

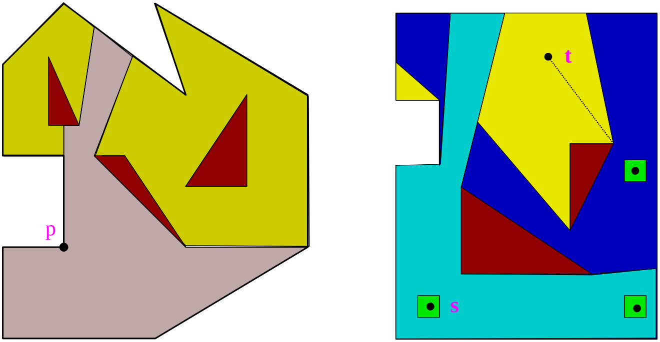


The visibility polygon of P from a point p (denoted as $VP(P,p)$) is the set of all points of P that are visible from p.

In other words, for every point $z \in P$, if the line segment joining z and p lies inside P, then z belongs to $VP(P,p)$.

1. S. K. Ghosh, *Visibility Algorithms in the Plane*, Cambridge University Press, United Kingdom, 2007.

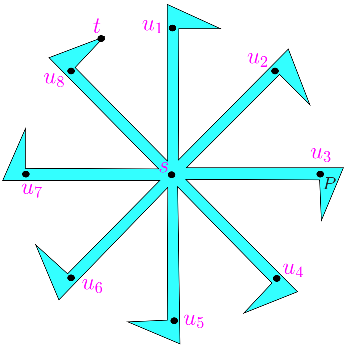
Continuous and discrete visibility



If the robot computes visibility polygons from each points on its path, we say that P is explored under continuous visibility.

If the robot computes visibility polygons from a selected set of points on its path, we say that P is explored under discrete visibility.

Target searching in a simple polygon with continuous visibility



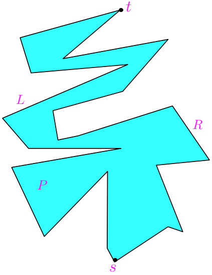
- Let $u_1, u_2, \ldots u_{n/4}$ be the nearest points of s in the alleys of a simple polygon P of distance d such that if the robot moves from s to u_i for each i, the robot can see the alley completely.
- In order to search t, the robot moves from s to u_i in each alley and then returns to s if it does not locate t.
- For every unsuccessful search, the robot travels $2d$ distance.

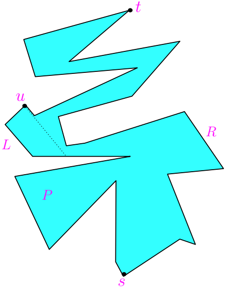
- In the worst case, the robot locates t in the last alley.
- So, the total distance travelled by the robot is at least $2d(n/4 - 1) + d$.
- Hence, the lower bound of the competitive ratio for this problem is $n/2 - 1$.

1. R. Klein, *Algorithmische Geometrie*, Second Edition, Springer-Verlag, 2005.
2. S. Schuierer, *On-line searching in simple polygons*, Proceeding of the International Workshop on Sensor Based Intelligent Robots, LNCS 1724, pp. 220–239, Springer-Verlag, 1999.

Competitive ratio: $2n - 7$.

Searching for a target in an unknown street





A simple polygon P is said to be a *street* (also called *LR-visibility polygon*) if there exists two points s and t on the boundary of P such that every point of the clockwise boundary from s to t of P (denoted as L) is visible from some point of the counterclockwise boundary of P from s to t (denoted as R) and vice versa.

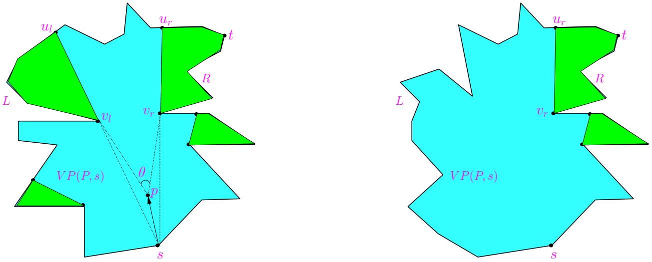
Observe that if a point robot moves along any path between s and t inside the street P, it can see all points of P.

Algorithms for target searching in an unknown street

1. R. Klein, *Walking an unknown street with bounded detour*, Computational Geometry: Theory and Applications, 1 (1992), 325-351. Competitive ratio: 5.72.
2. C. Icking, *Motion and visibility in simple polygons*, Ph.D. Thesis, FernUniversität, 1994. Competitive ratio: 4.44.
3. J. Kleinberg, *On line search in a simple polygon*, In Proceedings of the fifth ACM-SIAM Symposium on Discrete Algorithms, Pages 8-15, 1994. Competitive ratio: 2.61.
4. A. López-Ortiz and S. Schuierer, *Going home through an unknown street*, Proceedings of Algorithms and Data Structures, LNCS 955, pp. 135-146, Springer-Verlag, 1995. Competitive ratio: 2.05.
5. A. López-Ortiz and S. Schuierer, *Walking streets faster*, Proceedings of the 5th Scandinavian Workshop on Algorithm Theory, LNCS 1097,pp. 345-356, Springer-Verlag, 1996. Competitive ratio: 1.73.

6. P. Dasgupta and P. Chakrabarti and S. De Sarkar, *A new competitive algorithm for agent searching in unknown streets*, Proceeding of the 16th Symposium on FSTTCS, LNCS 1180, pp. 32-41, Springer-Verlag, 1995. Competitive ratio: 1.71.
7. I. Semrau, *Analyse und experimentelle Untersuchung von Strategien zum Finden eines Ziels in Str$\alpha\beta$enpolygonen*, Diploma Thesis, FernUniversität, 1996. Competitive ratio: 1.57.
8. E. Kranakis and A. Spatharis, *Almost optimal on-line search in unknown streets*, Proceedings of the 9th Canadian Conference on Computational Geometry, pp. 93-99, 1997. Competitive ratio: 1.498.
9. C. Icking, R. Klein, E. Langetepe and S. Schuierer, *An optimal competitive strategy for walking in streets*, SIAM Journal on Computing, 33(2004), 462-486. Competitive ratio: 1.41.

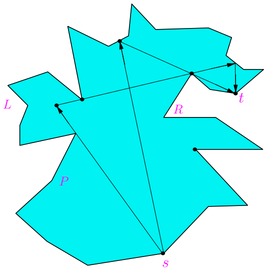
Optimal online algorithm for target searching in an unknown street



The left and right constructed edges of $VP(P,s)$ decide the movement of the robot initially. If $\theta < \pi/2$, then the robot follows the bisector of θ till it reaches a point where θ becomes $\pi/2$.

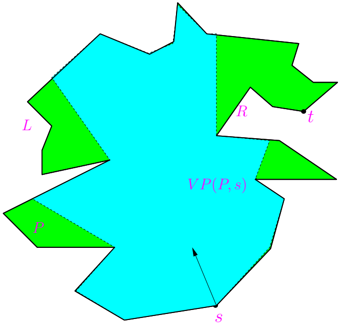
Then the robot follows a curve path toward $v_l v_r$ which is define by an algebraic expression based on positions of current p, v_l and v_r.

Optimal algorithm for target searching using link paths



Another problem for searching t in an unknown street P is find a path such that the number of links (or, turns) in the path is as small as possible.

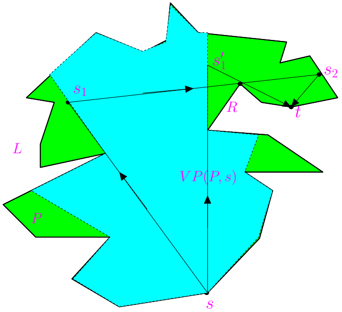
1. S. K. Ghosh and S. Saluja, *Optimal on-line algorithms for walking with minimum number of turns in unknown streets*, Computational Geometry: Theory and Applications, 8 (1997), 241-266. Competitive ratio: 2.



All right pockets occur before all left pockets while traversing the boundary of P in counterclockwise order from s.

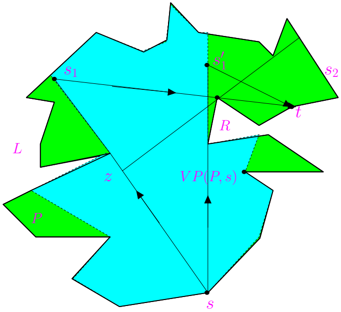
Observe that t belongs to either the leftmost top pocket or the rightmost top pocket.

If the robot takes any path within $VP(P, s)$ from s to a boundary point between the leftmost and rightmost pockets, it can see all points in every pocket except possibly one.

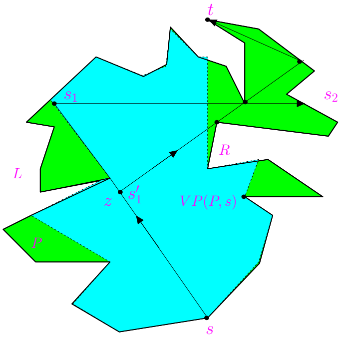


If the shortest path from s to t makes only right turns or only left turns, then $m+1$ links are sufficient for the robot to reach from s to t, where m is the link distance between s and t.

1. S. K. Ghosh, *Computing visibility polygon from a convex set and related problems*, Journal of Algorithms, 12(1991), 75-95.



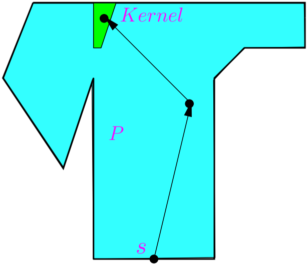
The robot has decided not to turn at z which turns out to be a correct decision as the shortest path from s to t makes only right turn.



The robot has decided not to turn at z as before but it is a wrong decision as the shortest path from s to t makes both types of turns. So, the robot backtracks to z and follows the correct path.

Since the robot takes one extra link for every such change in turn in the shortest path the robot takes at most $2m - 1$ links to reach from s to t. So, the competitive ratio of the online algorithm is $2 - 1/m$ which is shown to be optimal.

Walking into the kernel in an unknown star-shaped polygon with continuous visibility



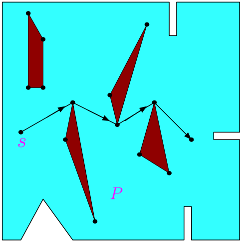
Starting from the initial position s, the problem is to design a competitive strategy to walk into the kernel of P.

1. C. Icking and R. Klein, *Searching for the Kernel of a Polygon—A Competitive Strategy*, SOCG, pages 258-266, 1995. Competitive ratio:5.331.

Algorithms for walking into the kernel

2. J.-H. Lee and K.-Y. Chwa, *Tight analysis of a self-approaching strategy for the online kernel-search problem*, Information Processing Letters, 69:39-45, 1999.
3. J.-H. Lee, C.-S. Shin, J.-H. Kim, S. Y. Shin and K.-Y. Chwa, *New competitive strategies for searching in unknown star-shaped polygons*, SOCG, pages 427-432, 1997. Competitive ratio: 3.828.
4. L. Palios, *A new competitive strategy for reaching the kernel of an unknown polygon*, Proceedings of 7th Workshop on Algorithmic Theory, LNCS 1851, pp. 367-382, Springer, 2000. Competitive ratio: 3.1226.
5. P. Anderson and A. Lopez-Ortiz, *A new lower bound for kernel searching*, CCCG, 2000. Lower bound: 1.515.
6. A. López-Ortiz and S. Schuierer, *Searching and on-line recognition of star-shaped polygons*, Information and Computations, 185:66-88, 2003. Lower bound: 1.5.

Exploring unknown polygons: continuous visibility

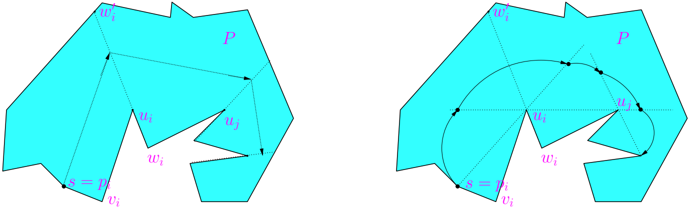


Starting from a point s inside P, the exploration problem is to design an online algorithm which a point robot can use for moving inside P such that every point of P becomes visible from some point on the exploration path of the robot

However, if P contains holes, the exploration problem does not admit competitive strategy.

1. X. Deng, T. Kameda and C. Papadimitriou, *How to learn an unknown environment*, Proceedings of the 32nd Annual IEEE Symposium on Foundation of Computer Science, PP. 298-303, 1991.

Exploring simple polygons: continuous visibility



Observe that if both edges of every reflex vertex u_i of P are seen by the robot, then the entire P has been explored by the robot

1. F. Hoffmann, C. Icking, R. Klein and K. Kriegel, *The polygon exploration problem*, SIAM Journal on Computing, 31:577-600, 2001. Competitive ratio: 26.5.

Exploring unknown polygons: discrete visibility

In the next part of the lecture, exploration algorithms and their competitive ratios are presented from the following papers on discrete visibility.

1. S. K. Ghosh, J. W. Burdick, A. Bhattacharya and S. Sarkar, *On-line algorithms with discrete visibility: Exploring unknown polygonal environments*, Special issue on Computational Geometry approaches in Path Planning, IEEE Robotics and Automation Magazine, vol. 15, no. 2, pp. 67-76, 2008.
2. S. K. Ghosh and J. W. Burdick, *An on-line algorithm for exploring an unknown polygonal environment by a point robot*, Proceedings of the 9th Canadian Conference on Computational Geometry, pp. 100-105, 1997.
3. A. Bhattacharya, S. K. Ghosh and S. Sarkar, *Exploring an Unknown Polygonal Environment with Bounded Visibility*,Proceedings of the International Conference on Computational Science, Lecture Notes in Computer Science, No. 2073, pp. 640-648, Springer Verlag, 2001.

Motivation for discrete visibility

Many on-line computational geometry algorithms for exploring unknown polygons assume that the visibility region can be determined in a continuous fashion from each point on a path of a robot. Is this assumption reasonable?

1. Autonomous robots can only carry a limited amount of on-board computing capability.
2. At the current state of the art, computer vision algorithms that could compute visibility polygons are time consuming.
3. The computing limitations suggest that it may not be practically feasible to continuously compute the visibility polygon along the robot's trajectory.
4. For good visibility, the robot's camera will typically be mounted on a mast and such devices vibrate during the robot's movement.
5. Hence for good precision the camera must be stationary while computing visibility polygons.

It seems feasible to compute visibility polygons only at a discrete number of points.

Exploration cost

Is the cost associated with a robot's physical movement dominate all other associated costs?

The essential components that contribute to the total cost required for a robotic exploration can be analyzed as follows. Each move will have two associated costs as follows.

1. 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 $r\ D$, where D is the total path length.
2. In an exploratory process where the robot has no apriori knowledge of the environment's geometry, each move must be planned immediately prior to the move so as to account for the most recently acquired geometric information. The robot will be stationary during this process, which we assume to take time t_M.
3. Since the robot is stationary during each sensing operation, we assume that it takes time t_S.

Let N_M and N_S be respectively the number of moves and the number of sensor operations required to complete the exploration of P. Hence, the total cost of an exploration is equated to the total time T required to explore P: $T(P) = t_M \; N_M + t_S \; N_S + r \; D$.

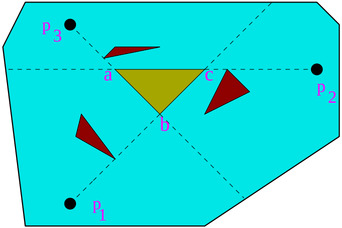
Now, $(t_M \; N_M + t_S \; N_S)$ can be viewed as the time required for computing and maintaining visibility polygons by computer vision algorithms, which is indeed a significant fraction of $T(P)$ because computer vision algorithms consume significant time on modest computers in a relatively cluttered environment.

Therefore, we assume that the overall cost of exploration is proportional to the cost for computing visibility polygons.

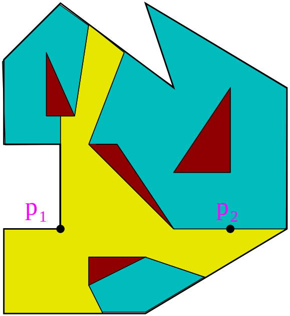
The criteria for minimizing the cost for robotic exploration is to reduce the number of visibility polygons that the on-line algorithms compute.

1. J. Borenstein and H. R. Everett and L. Feng, *Navigating mobile robots: sensors and techniques*, A. K. Peters Ltd., Wellesley, MA, 1995.
2. O. Faugeras, *Three-dimensional computer vision*, MIT Press, Cambridge, 1993.

An exploration algorithm



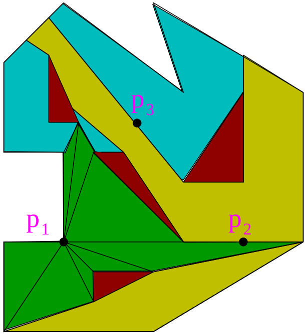
- We present an exploration algorithm that a point robot can use to explore an unknown polygonal environment P under discrete visibility.
- In order to explore P, the robot starts from a given position, and sees all points of the free space incrementally.
- It may appear that it is enough to see all vertices and edges of P in order to see the entire free-space. However, this is not the case.
- Three views from p_1, p_2 and p_3 are enough to see all vertices and edges of P but not the entire free-space of P.



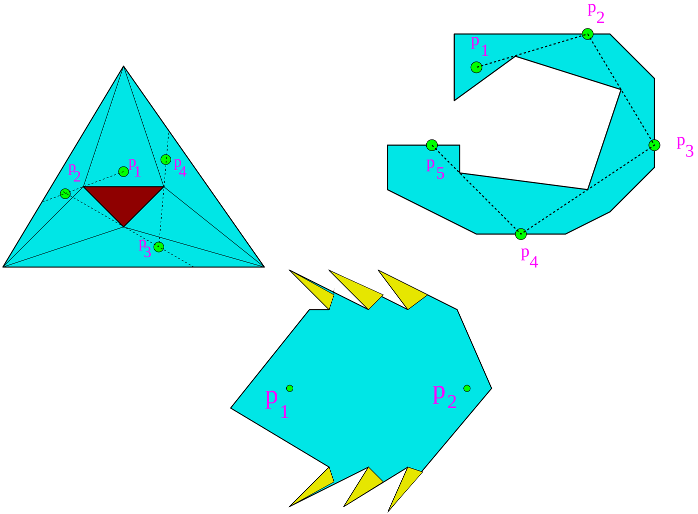
(i) Let S denote the set of viewing points that the algorithm has computed so far. (ii) The triangulation of P is denoted as $T(P)$. (iii) The visibility polygon of P from a point p_i is denoted as $VP(P, p_i)$.

Step 1: $i := 1$; $T(P) := \emptyset$; $S := \emptyset$; Let p_1 denote the starting position of the robot.
Step 2: Compute $VP(P, p_i)$; Construct the triangulation $T'(P)$ of $VP(P, p_i)$; $T(P) := T(P) \cup T'(P)$; $S = S \cup p_i$;
Step 3: *While* $VP(P, p_i) - T(P) = \emptyset$ *and* $i \neq 0$ *then* $i := i - 1$;

Step 4: *If* $i = 0$ *then goto* Step 7;
Step 5: *If* $VP(P, p_i) - T(P) \neq \emptyset$ *then* choose a point z on any constructed of $VP(P, p_i)$ lying outside $T(P)$;
Step 6: $i := i + 1$; $p_i := z$; *goto* Step 2;
Step 7: Output S and $T(P)$;
Step 8: Stop.



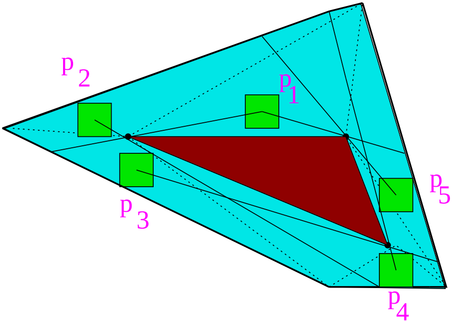
Competitive ratio



The algorithm needs $r+1$ views. Competitive ratio is $(r+1)/2$, where r denotes the number of reflex vertices of the polygon.

Open Problem: Can the bound be improved?

Convex robot exploration



We wish to design an algorithm that a convex robot C can use to explore an unknown polygonal environment P (under translation) following the similar strategy of a point robot.

C needs more than $r + 1$ views for exploration.

Open problem

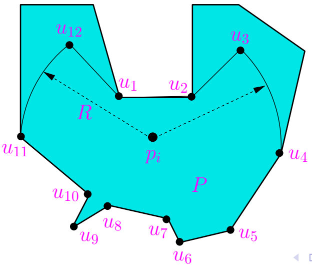
Can one derive an upper bound on the number of views for a convex robot exploration?

Exploring an unknown polygon: Bounded visibility

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.

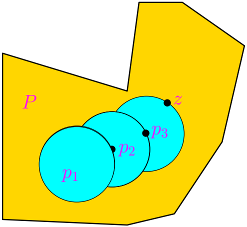
Therefore, the portion of the boundary of a polygonal environment within the range distance R is only considered to be visible from the camera of the robot.

Vertices of restricted visibility polygon from p_i with range R are $u_1, u_2, \ldots, u_{12}$.

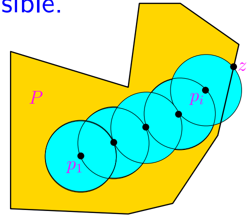


An exploration algorithm using restricted visibility

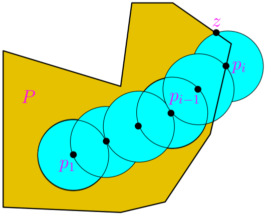
- The algorithm starts by computing the restricted visibility polygon $RVP(P, p_1)$ from the starting position p_1.



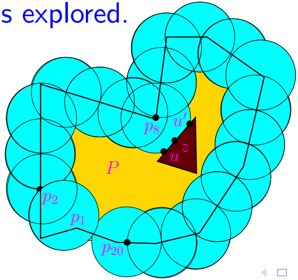
- It chooses the next viewing point p_i on a constructed edge or a circular edge of $RVP(P, p_{i-1})$ for $i \geq 1$ till a boundary point z of P becomes visible.



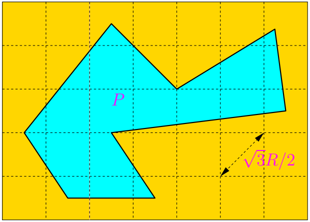
- Taking z as the next viewing point p_i, $RVP(P, p_i)$ is computed. Taking viewing points along the boundary of P in this fashion, restricted visibility polygons are computed till all points of this boundary of P become visible.



- The process of computing restricted visibility polygons ends once the entire P is explored.



Competitive ratio



The maximum number of views needed to explore the unknown polygon P with h obstacles of size n is bounded by

$$\left\lfloor \frac{8 \times Area(P)}{3 \times R^2} \right\rfloor + \left\lfloor \frac{Perimeter(P)}{R} \right\rfloor + r + h + 1.$$

The competitive ratio of the algorithm is

$$\left\lfloor \frac{8\pi}{3} + \frac{\pi R \times Perimeter(P)}{Area(P)} + \frac{(r+h+1) \times \pi R^2}{Area(P)} \right\rfloor.$$

Open problem

Can one improve the competitive ratio of the algorithm?

Exploration and Coverage Algorithms

1. A. Bhattacharya, S. K. Ghosh and S. Sarkar, *Exploring an Unknown Polygonal Environment with Bounded Visibility*, Lecture Notes in Computer Science, No. 2073, pp. 640-648, Springer Verlag, 2001.
2. S. K. Ghosh, J. W. Burdick, A. Bhattacharya and S. Sarkar, *On-line algorithms for exploring unknown polygonal environments with discrete visibility*, Special issue on Computational Geometry approaches in Path Planning, IEEE Robotics and Automation Magazine, 2008 vol. 15, no. 2, pp. 67-76, 2008.
3. E. U. Acar and H. Choset, *Sensor-based coverage of unknown environments: Incremental construction of morse decompositions*, The International Journal of Robotics Research, 21 (2002), 345-366.
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 (1993), 227-244.

5. H. Choset, *Coverage for robotics– A survey of recent results*, Annals of Mathematics and Artificial Intelligence, 31 (2001), 113-126.
6. X. Deng, T. Kameda and C. Papadimitriou, *How to learn an unknown environment I: The rectilinear case*, Journal of ACM, 45 (1998), 215-245.
7. F. Hoffmann, C. Icking, R. Klein and K. Kriegel, *The polygon exploration problem*, SIAM Journal on Computing, 31 (2001), 577-600.
8. C.J. Taylor and D.J. Kriegman, *Vison-based motion planning and exploration algorithms for mobile robot*, IEEE Transaction on Robotics and Automation, 14 (1998), 417-426.
9. P. Wang, *View planning with combined view and travel cost*, Ph. D. Thesis, Simon Fraser University, Canada, 2007.
10. S. P. Fekete and C. Schmidt, *Polygon exploration with time-discrete vision*, Computational Geometry: Theory and Applications, 43:148-168, 2010.

Simple robots

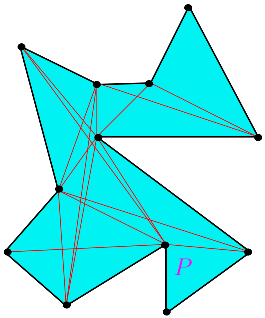
- So far, we have considered autonomous mobile robots that have best capabilities for movement, sensing, computation and communications.
- Naturally, such sophisticated robots are costly and deploying a large number of them for a given task may be impractical.
- On the other hand, there are tasks that do not require sophisticated robots. For example, many cleaning robots depend on contact sensors only, and make somewhat random movements rather than using sophisticated hardware for optimizing their trajectories.
- Such simple design robots in terms of hardware are cheap and are, therefore, suitable for mass market.
- The question is: What capabilities does a robot need at the very least for a given task?

Mobile agents

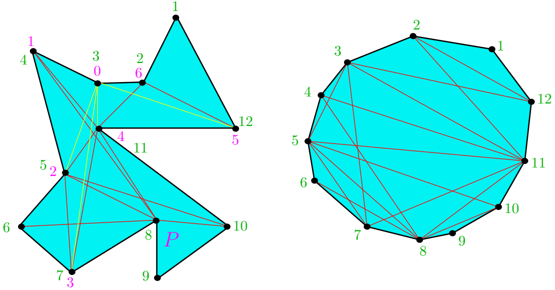
- What capabilities does a robot need at the very least for exploring an unknown polygon?
- Is it enough for a robot to produce a rough map of the polygon?
- To address these questions, we use a very basic theoretical robot model with additional atomic capabilities.
- Results from such theoretical models can provide a reference for a realistic design.
- These theoretical robots are referred as *Agents* rather that (realistic) robots.

1. S. Suri, E. Vicari and P. Widmayer, *Simple robots with minimal sensing: From local visibility to global geometry*, International Journal on Robotic Research, 27(9):1055-1067, 2008.
2. Y. Disser, *Mapping polygons*, Ph.D. thesis, ETH Zurich, 2011.

Mapping polygons using mobile agents



- Unlike normal movements of a robot, agents are restricted to move only along the edges of the visibility graph of the polygon P.
- The visibility graph of P is a graph whose vertex set consists of the vertices of P and whose edges are visible pairs of vertices of P.
- This means that an agent moves from a vertex to another vertex inside P along the lines of sights.

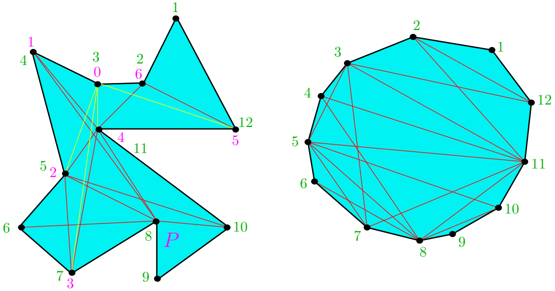


- While located at a vertex, an agent can use its sensor to locate the vertices of P visible from the current position in the counter-clockwise order along the boundary of P.
- However, the agent neither can provide co-ordinates of these visible vertices nor knows the polygonal numbering of these visible vertices.
- Moreover, the agent cannot recognize vertices that are seen earlier from other vertices.
- After exploration, the agent outputs the visibility graph of P as a rough map of P.

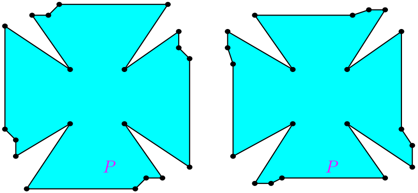
Exploration strategy

- Exploration strategy consists of data collection phase and computation phase.
- Data collection phase is always the same: the agent traverses the boundary of P and stops at each vertex for collecting available data including the list of visible vertices.
- The computation phase does not involve any further exploration of P, and in this phase, computation is carried out for constructing the visibility graph of P.
- Can visibility graph of P be constructed always from available data?

Boundary traversal

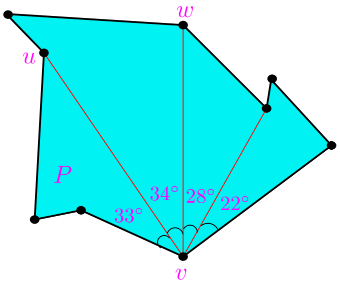


- Starting from a vertex, an agent can traverse the boundary of P in counter-clockwise order by following the first counter-clockwise visible edge from the current position.
- If the agent can distinguish a vertex from all other vertices of P, then the visibility graph of P can be constructed easily.
- If the visibility graph is not symmetric, then there is a good chance to locate a vertex that can be distinguished for all other vertices.



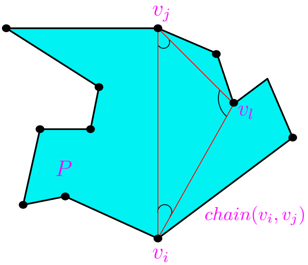
- Suppose an agent knows the total number of vertices n of P before the boundary traversal. It also has an additional capability to measure the angle at each vertex of P.
- Even with these enhanced capabilities, the agent cannot always reconstruct the visibility graph of P.

1. D. Bilò, Y. Disser, M. Mihalák, S. Suri, E. Vicari and P. Widmayer, *Reconstructing visibility graphs with simple robots*, Theoretical Computer Science, 444:52-59, 2012.
2. Y. Disser, D. Bilò, M. Mihalák, S. Suri, E. Vicari and P. Widmayer, *On the limitations of combinatorial visibilities*, Proc. of the 25th European Workshop on Computational Geometry, pp. 207-210, 2009.



- Let u and w be two consecutive visible vertices in the angular order of any vertex v.
- Suppose, the capability of an basic agent is enhanced with an angle sensor such that the agent can measure the exact angle between (v, u) and (v, w) at v for all v, u and w in P.
- Such agents can always construct the visibility graph of a simple polygon P with or without the prior knowledge of n.

1. Y. Disser, M. Mihalák and P. Widmayer, *A Polygon is Determined by its Angles*, Computational Geometry: Theory and Applications, 44:418-426, 2011.



- Assume that all visible edges connecting vertices of $chain(v_i, v_j)$ are identified except the edge (v_i, v_j), and the algorithm wants to determine whether (v_i, v_j) is a visible edge.
- Consider a vertex $v_l \in chain(v_i, v_j)$ such that (v_i, v_l) and (v_l, v_j) are visible edges.
- If no such vertex v_l exists, then obviously (v_i, v_j) is not a visible edge.
- If the internal angles at v_i and v_j of the triangle (v_i, v_l, v_j) match with the corresponding measured angles at v_i and v_j by the agent, then (v_i, v_j) is a visible edge.

- By testing every pair of vertices (v_i, v_j) of P with distances $2, 3, 4, \ldots$, all pairs of visible vertices of P can be identified.
- In addition, a simple polygon P' can also be reconstructed from the visibility graph of P using the measured angles.
- The overall running time of the algorithm can be improved from $O(n^3 \log n)$ to $O(n^2)$.

1. D. Z. Chen and H. Wang, *An improved algorithm for reconstructing a simple polygon from its visibility angles*, Computational Geometry: Theory and Applications 45:254-257, 2012.

Agents with other type sensors

- The angle-type sensor allows to determine whether an angle is convex or reflex.
- A look-back sensor allows an agent to return to its previous position, i.e., if an agent moves from a vertex u to another vertex v, the agent can return to u using its look-back sensor.
- A compass allows an agent to measure the angle at each edge with respect to the global directions.
- Several algorithms have been proposed for constructing visibility graphs of unknown simple polygons using different combination of sensors.

1. J. Chalopin, S. Das, Y. Disser, M. Mihalák and P. Widmayer, *Telling Convex from Reflex Allows to Map a Polygon*, Proc. of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS), pp. 153-164, 2011.
2. J. Chalopin, S. Das, Y. Disser, M. Mihalák and P. Widmayer, *Mapping simple polygons: How robots benefit from looking back*, Algorithmica, 2012 (to appear).

Mapping polygons with holes

- If P is a polygon with holes, the problem of constructing visibility graphs of P becomes a much a harder problem.
- Recently, an exploration algorithm has been design for exploring such polygons using basic agents having compass with a prior knowledge of an upper bound on the number of vertices of P.

1. Y. Disser, M. Mihalák, S. K. Ghosh and P. Widmayer, *Mapping a polygon with holes using compass*, Proc. of the 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, 2012 (to appear).

Open problems

There are several open problems for constructing visibility graphs of unknown polygons P with or without holes for boundary traversal as well for unrestricted traversal of mobile agents with or without additional capabilities.

1. Y. Disser, *Mapping polygons*, Ph.D. thesis, ETH Zurich, 2011.

Thank You.