Visibility-based Robot Path Planning

Subir Kumar Ghosh

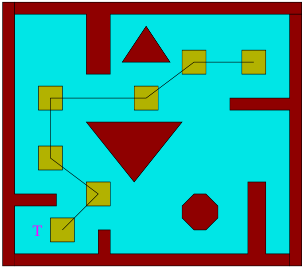
School of Technology & Computer Science
Tata Institute of Fundamental Research
Mumbai 400005, India
ghosh@tifr.res.in

Overview

1. Computing the configuration space
2. Computing Euclidean shortest paths
3. Computing minimum link paths
4. Computing bounded curvature paths
5. Exploring an unknown polygon: Continuous visibility
6. Exploring an unknown polygon: Discrete visibility
7. Exploring an unknown polygon: Bounded visibility

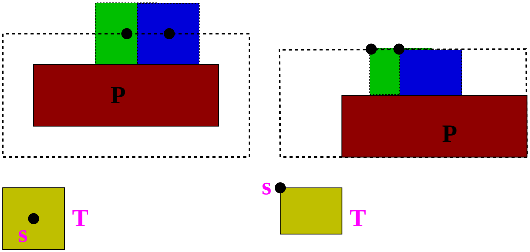
Collision-free Path

One of the main problems in robotics, called robot path planning, is to find a collision-free path amidst obstacles for a robot from its starting position to its destination.



1. J-C Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.
2. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.

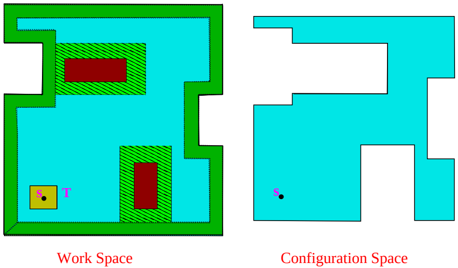
Minkowski sum



Above figures show the Minkowski sums of P and T with s as the reference point (under translation).

1. M. de Berg, M. Van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer, 1997.
2. P.K. Ghosh, *A solution of polygon containment, spatial planning, and other related problems using Minkowski operations*, Computer Vision, Graphics and Image Processing, 49 (1990), 1-35.
3. P.K. Ghosh, *A unified computational framework for Minkowski operations*, Computers and Graphics, 17 (1993), 357-378.

Computing configuration space



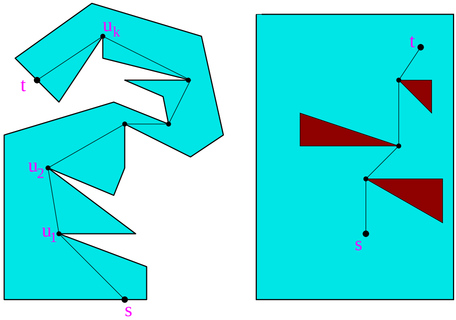
The configuration space can be computed using Minkowski sum.

The problem of computing collision-free path of a rectangle in the actual space is now reduced to that of a point in the free configuration space.

1. T. Lozano-Perez and M. A. Wesley, *An algorithm for planning collision-free paths among polyhedral obstacles*, Communication of ACM, 22 (1979), 560-570.

Computing Euclidean shortest paths

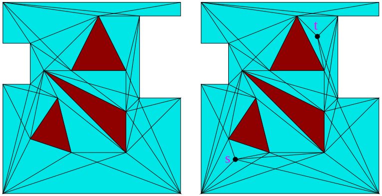
The *Euclidean shortest path* (denoted as $SP(s, t)$) between two points s and t in a polygon P is the path of smallest length between s and t lying totally inside P.



Let $SP(s, t) = (s,\ u_1,\ u_2, \ldots,\ u_k,\ t)$. Then, (i) $SP(s, t)$ is a simple path, (ii) u_1, u_2,..., u_k are vertices of P and (iii) for all i, u_i and u_{i+1} are mutually visible in P. $SP(s, t)$ is *outward convex* at every vertex on the path.

Computing $SP(s,t)$ using visibility graph

The visibility graph of a polygon P with polygonal holes or obstacles is a graph whose vertex set consists of the vertices of P and whose edges are visible pairs of vertices.



Assign the length of each visible pair as an weight on the corresponding edge in the visibility graph and use the following algorithm to compute $SP(s,t)$.

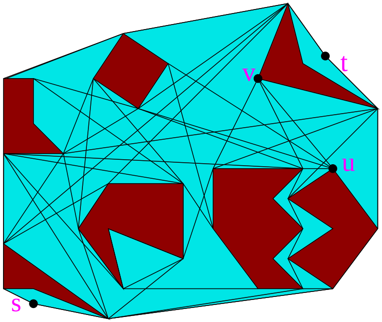
1. M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, Journal of ACM, 34 (1987), 596-615. Running time: $O(n \log n + E)$, where E is the number of edges in the visibility graph.

Algorithms for computing visibility graphs

1. T. Lozano-Perez and M. A. Wesley, *An algorithm for planning collision-free paths among polyhedral obstacles*, Communication of ACM, 22 (1979), 560-570. Running time: $O(n^3)$.
2. D. T. Lee, *Proximity and reachability in the plane*, Ph.D. Thesis, University of Illinois, 1978. Running time: $O(n^2 \log n)$.
3. M. Sharir and A. Schorr, *On shortest paths in polyhedral spaces*, SIAM Journal on Computing, 15 (1986), 193-215. Running time: $O(n^2 \log n)$.
4. E. Welzl, *Constructing the visibility graph for n line segments in $O(n^2)$ time*, Information Processing Letters, 20 (1985), 167-171. Running time: $O(n^2)$.
5. T. Asano and T. Asano and L. J. Guibas and J. Hershberger and H. Imai, *Visibility of disjoint polygons*, Algorithmica, 1 (1986), 49-63. Running time: $O(n^2)$.

6. M. Overmars and E. Welzl, *New methods for constructing visibility graphs*, in Proc. 4th ACM Symposium on Computational Geometry, 164-171, 1988. Running time: $O(E \log n)$, where E is the number of edges in the visibility graph.
7. S. K. Ghosh and D. M. Mount, *An output sensitive algorithm for computing visibility graphs*, SIAM Journal on Computing, 20 (1991), 888-910. Running time: $O(n \log n + E)$. Space: $O(E)$.
8. M. Pocchiola and G. Vegter, *Topologically sweeping visibility complexes via pseudo-triangulations*, Discrete and Computational Geometry, 16(1996), 419–453. Running time: $O(n \log n + E)$. Space: $O(n)$.
9. S. Kapoor and S. N. Maheshwari, *Efficiently Constructing the Visibility Graph of a Simple Polygon with Obstacles*, SIAM Journal on Computing, 30(2000), 847-871. Running time: $O(h \log n + T + E)$, where T is the time for triangulation and h is the number of holes.

Computing $SP(s,t)$ using partial visibility graph



Since $SP(s,t)$ is outward convex at every vertex on the path, it is enough to consider only those edges (u,v) of the visibility graph that are tangential at u and v.

Algorithms for computing partial visibility graphs

1. S. Kapoor, S. N. Maheshwari and J. Mitchell, *An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane*, Discrete and Computational Geometry, 18(1997), 377-383. Running time: $O(n + h^2 \log n)$, where h is the number of holes. Space: $O(n)$.
2. H. Rohnert, *Shortest paths in the plane with convex polygonal obstacles*, Information Processing Letters, 23 (1986), 71-76. Running time: $O(n \log n + h^2)$.

Review Articles

1. J. Mitchell, *Geometric shortest path and network optimization*, Handbook in Computational Geometry (edited by J.-R Sack and J. Urrutia), Elsevier Science Publishers B.W., Chapter 15, pp. 633-702, 2000.
2. T. Asano, S. K. Ghosh and T. C. Shermer. Visibility in the plane. *Handbook in Computational Geometry (ed. J.-R. Sack and J. Urruta)*, Elsevier Science Publishers B. W., Chapter 19, pp. 829-876, 2000.

Computing $SP(s,t)$ directly

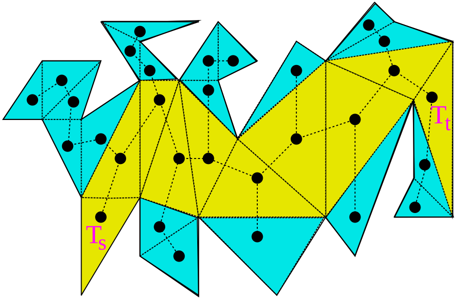
$SP(s,t)$ can be computed directly once the shortest path map is constructed using continuous Dijkstra Method.
This method involves simulating the effect of a 'wavefront' propagating out of s. The *wavefront* at a distance d from s is the set of all points $u \in P$ such that $|SP(s,u)| = d$.
The propagation can be carried out on *cell-by-cell* basis after decomposing the entire region of P into "conforming subdivision".

Open Problem

Can $SP(s,t)$ be computed in $O(n + h \log h)$ time and $O(n)$ space?

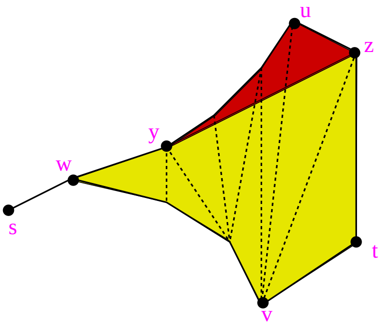
1. J. Hershberger and S. Suri, *An optimal-time algorithm for Euclidean shortest paths in the plane*, SIAM Journal on Computing, 28(1999), 2215-2256. Running time: $O(n \log n)$. Space: $O(n \log n)$.

Computing $SP(s, t)$ in a simple polygon



The dual graph of a triangulation of a simple polygon is a tree. $SP(s, t)$ passes only through the triangles in the path from T_s and T_t in the dual tree.

1. B. Chazelle. *Triangulating a simple polygon in linear time*, Discrete and Computational Geometry, 6(1991), 485-529. Running time: $O(n)$.



Let (u, v, z) be a triangle such that $SP(s, u)$ and $SP(s, v)$ have already been computed. Then $SP(s, z)$ can be computed by drawing tangent from z to $SP(s, u)$ or $SP(s, v)$.

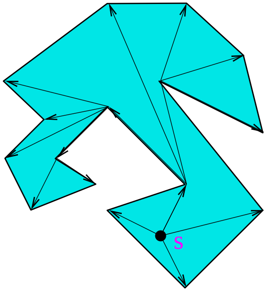
Open Problem

Can $SP(s, t)$ be computed in a simple polygon in $O(n)$ time without triangulation?

1. D. Lee and F. Preparata, *Euclidean shortest paths in the presence of rectilinear boundaries*, Networks, 14 (1984), 303-410. Running time: $O(n)$.

Computing the Euclidean shortest path tree

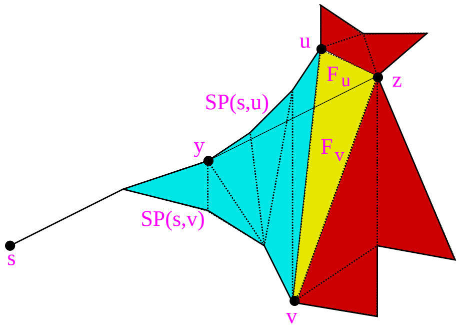
The Euclidean shortest path tree from s is the union of $SP(s, u)$ to all vertices u of the polygon.



The figure shows the Euclidean shortest path tree from a given point s to all vertices of a simple polygon.

1. L. Guibas, J. Hershberger, D. Leven, M. Sharir and R. Tarjan, *Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons*, *Algorithmica*, 2(1987), 209-233.

Running time: $O(n)$.

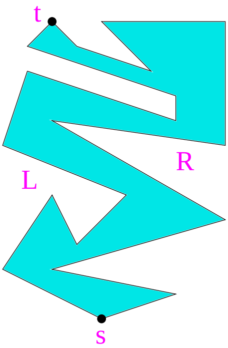


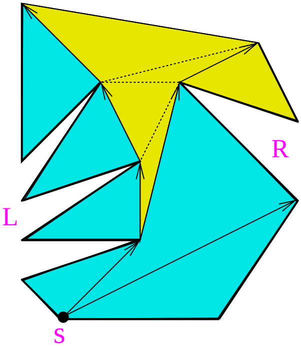
Tangent yz splits a funnel into two funnels and both funnels can be propagated in $O(n)$ time using Finger search tree.

Open Problems

1. Can the shortest path tree be computed from a point in a triangulated simple polygon in $O(n)$ time without using Finger search trees?
2. Can the shortest path tree be computed from a point in a simple polygon in $O(n)$ time without triangulation?

Computing shortest path tree without triangulation





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

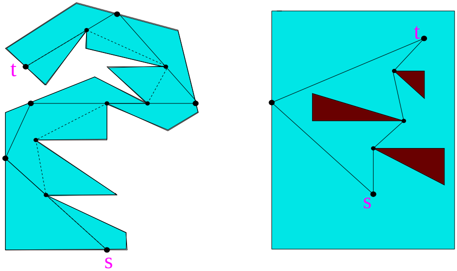
The shortest path tree from a point s inside a LR-visibility polygon P can be computed in $O(n)$ only by scanning the boundary of P, which also gives a triangulation of P.

Open Problem

Can a simple polygon be decomposed into LR-visibility polygons in $O(n)$ time?

1. P. J. Heffernan, *An optimal algorithm for the two-guard problems*, International Journal of Computational Geometry and Applications, 6 (1996), 15-44. Running time: $O(n)$.
2. G. Das, P. J. Heffernan and G. Narasimhan, *LR-visibility in polygons*, Computational Geometry: Theory and Applications, 7 (1997), 37-57. Running time: $O(n)$.
3. B. Bhattacharya and S. K. Ghosh, *Characterizing LR-visibility polygons and related problems*, Computational Geometry: Theory and Applications, 18 (2001), 19-36. Running time: $O(n)$.

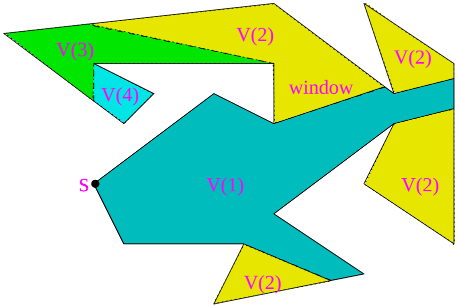
Computing minimum link paths



A minimum link path connecting two points s and t inside a polygon P with or without holes (denoted by $MLP(s,t)$) is a polygonal path with the smallest number of turns or links.

1. S. Suri, *A linear time algorithm for minimum link paths inside a simple polygon*, Computer Graphics, Vision, and Image Processing, 35 (1986), 99-110. Running time: $O(n)$.
2. S. K. Ghosh, *Computing the visibility polygon from a convex set and related problems*, Journal of Algorithms, 12 (1991), 75-95. Running time: $O(n)$.

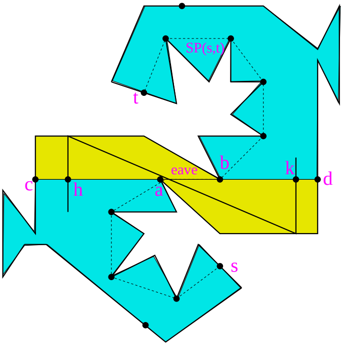
Suri's algorithm



$V(1)$ is the visibility polygon of s. For $i > 1$, $V(i)$ is the set of points of P weakly visible from some window of $V(i-1)$. Number of links (called *link distance*) required from s to any point of $V(i)$ is i. The turning points of a link path are on the windows.

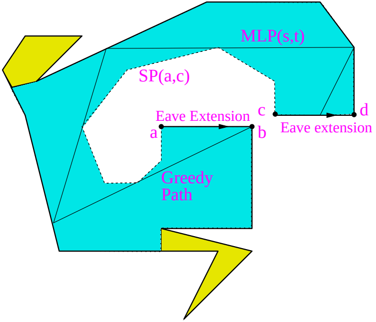
1. J. Hershberger, *An optimal visibility graph algorithm for triangulated simple polygons*, Algorithmica, 4 (1989), pp. 141-155. Running time: $O(E)$, where E is the number of edges in the visibility graph.

Ghosh's algorithm



Ghosh's algorithm transforms $SP(s,t)$ into $MLP(s,t)$:

- There exists a $MLP(s,t)$ containing all eaves of $SP(s,t)$.
- Compute MLP between the extensions of consecutive eaves and connected them using the subsegment containing eaves to construct $MLP(s,t)$.



- *MLP* between the extensions of consecutive eaves *ab* and *cd* is a convex greedy path inside the complete visibility polygon of P from $SP(a, c)$.

1. V. Chandru, S. K. Ghosh, A. Maheshwari, V. T. Rajan and S. Saluja, *NC-Algorithms for minimum link path and related problems*, Journal of Algorithms, 19 (1995), 173-203. Running time: $O(\log n \log\log n)$ with $O(n)$ processors in CREW PRAM model of computing.

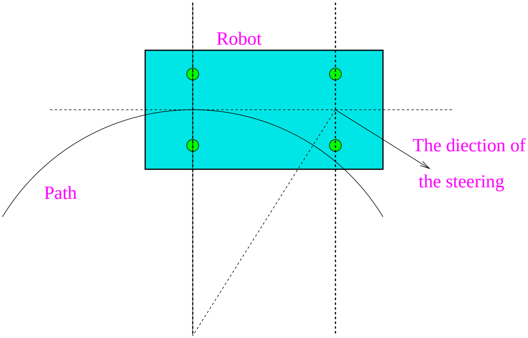
Computing $MLP(s,t)$ in a polygon with holes

The algorithm of Mitchell et al. for computing $MLP(s,t)$ in a polygon with holes follows the same approach as that of Suri by computing the regions $V(1)$, $V(2)$,...Since computing $V(i)$ explicitly, for all i, is very costly, algorithm computes only the envelope of $V(i)$ for all i which is enough to compute $MLP(s,t)$.

Open Problem: Can $MLP(s,t)$ be computed in a polygon with holes in sub-quadratic time?

1. J. Mitchell and G. Rote and G. Woeginger, *Minimum-link paths among obstacles in the plane*, Algorithmica, 8 (1992), 431-459. Running time: $O(Ea(n)\log^2 n)$ where $a(n)$ is the inverse of the Ackermann function.
2. A. Maheshwari, J-R. Sack and H. Djidjev, *Link distance problems*, Handbook in Computational Geometry (edited by J.-R Sack and J. Urrutia), Elsevier Science Publishers B.W., Chapter 12, pp. 519-558, 2000.
3. S. K. Ghosh, *Visibility Algorithms in the Plane*, Cambridge University Press, 2007.

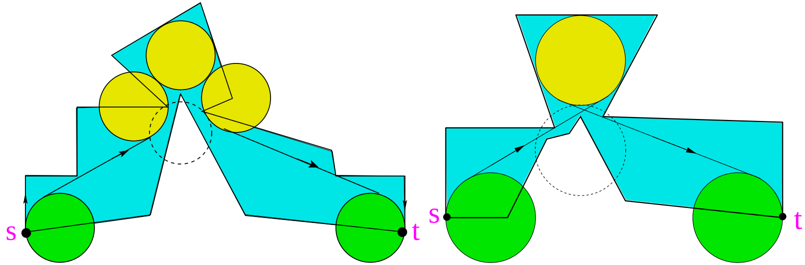
Non-holonomic Robot Motion Planning



A robot is said to be *non-holonomic* if some kinematics constraints (for example, velocity/acceleration bounds, curvature bounds) locally restricts the authorized directions for its velocity.

A typical example of a non-holonomic robot is that of a car: assuming no slipping of the wheels on the ground, the velocity of the midpoint between the two rear wheels of the car is always tangential to the path.

Bounded curvature path problem



Compute a path of minimum length inside a polygon between two given points s to t consisting of straight-line segments and circular arcs such that

(i) the radius of each circular arc is at least 1,

(ii) each segment on the path is the tangent between the two consecutive circular arcs on the path,

(iii) the given initial direction at s is tangent to the path at s,

(iv) the given final direction at t is tangent to the path at t.

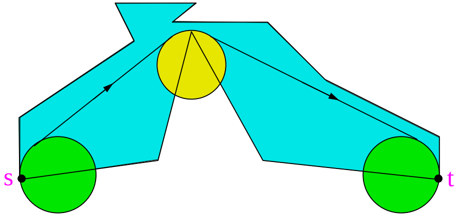
Open Problem: The above problem is open except when the given polygon is a convex polygon without holes.

Algorithms for bounded curvature paths

1. S. Fortune and G. Wilfong, *Planning constrained motion*, Proceedings of the 20th Annual ACM Symposium on Theory of Computing, pp. 445-459, 1988.
2. P. Jacobs and J. Canny, *Planning smooth paths for mobile robots*, Proceedings of the IEEE Conference on Robotics and Automation, pp. 2-7, 1989.
3. P.K. Agarwal, P. Raghavan, and H. Tamaki, *Motion Planning for a steering-constrained robot through moderate obstacles*, Proceedings of the 27th Annual ACM Symposium on Theory of Computing, pp. 343-352, 1995.
4. J-D. Boissonnat and S. Lazard, *A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles*, Proceedings of the Annual ACM Symposium on Computational Geometry, pp. 242-251, 1996.

5. P.K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides, *Curvature-constrained shortest paths in a convex polygon*, Proceedings of the 14th Annual ACM Symposium on Computational Geometry, pp. 392-401, 1998. Running time: $O(n^2 \log n)$.
6. J-D. Boissonnat, S. K. Ghosh, T. Kavitha and S. Lazard, *An algorithm for computing a convex and simple path of bounded curvature in a simple polygon*, Algorithmica 34 (2002), 109-156. Running time: $O(n^4)$.
7. J. Backer and D. Kirkpatrick, *Curvature-bounded traversals of narrow corridors*, Proceedings of the 21st Annual ACM Symposium on Computational Geometry, pp. 278-287, 2005.
8. J. Backer and D. Kirkpatrick, *Finding curvature-constrained paths that avoid polygonal obstacles*, Proceedings of the 21st Annual ACM Symposium on Computational Geometry, pp. 66-73, 2007.

Computing a convex and simple path

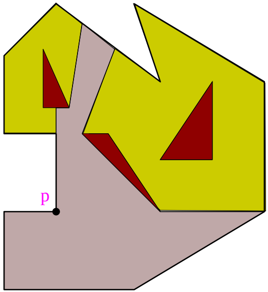


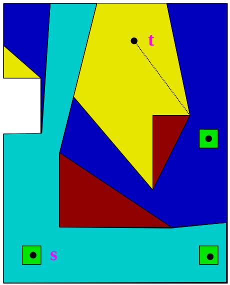
By constructing the locus of center of a circle of unit radius translating along the boundary of complete visibility polygon of P', the algorithm constructs a convex and simple path of bounded curvature in $O(n^4)$ time.

This algorithm is based on the relationship between the Euclidean shortest path, link paths and paths of bounded curvature.

Based on two new necessary conditions, a convex and simple path of bounded curvature can be constructed in $O(n^4)$ time whose length, except in special situations, is at most twice the optimal.

Exploring an unknown polygon: Continuous visibility





Suppose the polygon P is not known apriori and the point robot can compute the visibility polygon of P from its current position using visual sensors.

The robot wants to see all points of P with minimum cost. Cost can be the length or the number of links in the path that the robot has traveled starting from its initial position.

Efficiency of the on-line algorithm

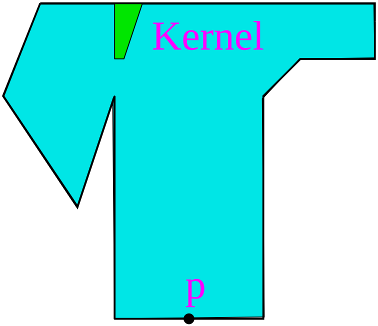
$$Competitive\ ratio = \frac{cost\ \ of\ \ the\ on - line\ algorithm}{cost\ \ of\ \ the\ off - line\ algorithm}$$

.

1. A. Blum and P. Raghavan and B. Schieber, *Navigating in unfamiliar geometric terrain*, SIAM Journal on Computing, 26 (1997), 110-137.
2. 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.
3. X. Deng and T. Kameda and C. Papadimitriou, *How to learn an unknown environment I: The rectilinear case*, Journal of ACM, 45 (1998), 215-245.

4. F. Hoffmann, C. Icking, R. Klein, K. Kriegel, *A competitive strategy for learning a polygon*, In Proceedings of the eighth ACM-SIAM Symposium on Discrete Algorithms, Pages 166-174, 1997.
5. F. Hoffmann, C. Icking, R. Klein and Klaus Kriegel, *The polygon exploration problem*, SIAM Journal on Computing, 31 (2001), 577-600.
6. A. Lopez-Ortiz and S. Schuierer, *Searching and on-line recognition of star-shaped polygons*, Information and Computations, 185(2003), 66-88.

Searching for the kernel



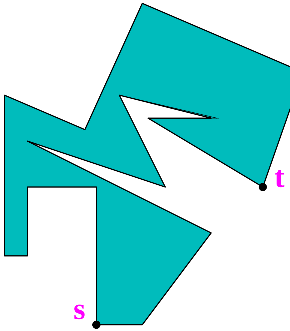
Starting from the intial position p, the problem is to design a competitive strategy to walk into the kernel of P.

Open Problem: The problem is open in link metric.

Kernel Searching Algorithms

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.
2. 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.
3. P. Anderson and A. Lopez-Ortiz, *A new lower bound for kernel searching*, CCCG, 2000. Competitive ratio: 1.515.

Searching for a target in a street



A *street* (also called *LR-polygon*) is a polygon for which two boundary chains from start to target are mutually weakly visible. So, the entire street is visible from any path from s to t.

The problem is to find a path from s to t such that the competitive ratio is the minimum.

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

Algorithms for searching a 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. 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: 2.83.
3. 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.
4. A. Lopez-Ortiz and S. Schuierer, *Lower bounds for streets and generalized streets*, International Journal of Computational Geometry and Applications, 11(2001), 401-421. Lower bounds: 1.41 and 9.06.
5. A. Datta and C. Icking, *Competitive searching in a generalized street*, CGTA, 13 (1999), 109-120. Competitive ratio: 9.06.
6. S. K. Ghosh and S. Saluja, *Optimal on-line algorithms for walking with minimum number of turns in unknown streets*, CGTA, 8 (1997), 241-266. Competitive ratio: 2.

Exploring an unknown polygon: 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?

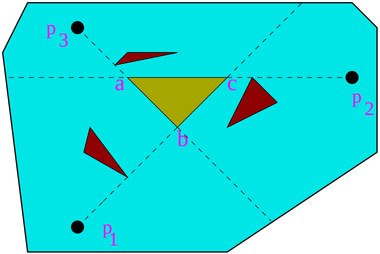
1. Autonomous robots can only carry a limited amount of on-board computing capability. At the current state of the art, computer vision algorithms that could compute visibility polygons are time consuming. The computing limitations suggest that it may not be practically feasible to continuously compute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

It seems feasible to compute visibility polygons only at a discrete number of points. Is the cost associated with a robot's physical movement dominate all other associated costs?

The criteria for minimizing the cost for robotic exploration is to reduce the number of visibility polygons.

Exploration under discrete visibility

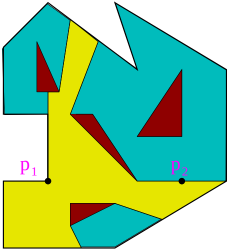
We wish to design an algorithm that a point robot can use to explore an unknown polygonal environment P under discrete visibility.



Three views are enough to see all vertices and edges of the polygon but not the entire free-space.

1. 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.

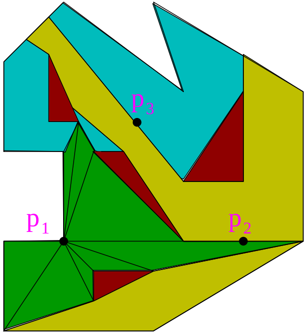
Exploration algorithm of Ghosh and Burdick



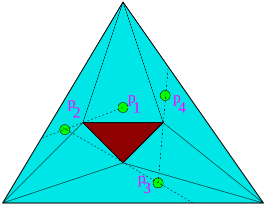
(i) Let S denote the set of viewing points that the algorithm has computed so far. (ii) The triangulation of P is denoted as $T(P)$. (iii) The visibility polygon of P from a point p_i is denoted as $VP(P, p_i)$.

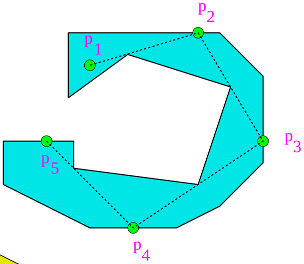
Step 1: $i := 1$; $T(P) := \emptyset$; $S := \emptyset$; Let p_1 denote the starting position of the robot.
Step 2: Compute $VP(P, p_i)$; Construct the triangulation $T'(P)$ of $VP(P, p_i)$; $T(P) := T(P) \cup T'(P)$; $S = S \cup p_i$;

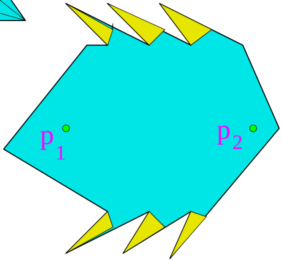
Step 3: *While* $VP(P, p_i) - T(P) = \emptyset$ *and* $i \neq 0$ *then* $i := i - 1$;
Step 4: *If* $i = 0$ *then goto* Step 7;
Step 5: *If* $VP(P, p_i) - T(P) \neq \emptyset$ *then* choose a point z on any constructed of $VP(P, p_i)$ lying outside $T(P)$;
Step 6: $i := i + 1$; $p_i := z$; *goto* Step 2;
Step 7: Output S and $T(P)$;
Step 8: Stop.



Competitive ratio







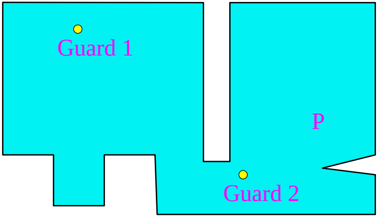
The algorithm needs $r+1$ views. Competitive ratio is $(r+1)/2$, where r denotes the number of reflex vertices of the polygon.

Open Problem: Can the bound be improved?

The art gallery problem

An art gallery can be viewed as a polygon P with or without holes with a total of n vertices and guards as points in P.

Victor Klee asked in 1976: How many guards are always sufficient to guard any polygon with n vertices?



The minimum vertex, point and edge guard problems for polygons with or without holes (including orthogonal polygons) are NP-hard.

1. J. O'Rourke, *Art gallery theorems and algorithms*, Oxford University Press, 1987.
2. J. Urrutia, Art Gallery and illumination problems, Handbook of Computational Geometry (Ed. J.-R. Sack and J. Urrutia), Elsevier Science, pp. 973-1027, 2000.

Approximation algorithms

1. S. K. Ghosh, Approximation algorithm for art gallery problems, Proceedings of the Canadian Information Processing Society Congress, pp. 429-434, 1987. Running time: $O(n^5 \log n)$ time. Approximation ratio: $O(\log n)$. Recently, the running time has been improved to $O(n^4)$ for simple polygons and $O(n^5)$ for polygons with holes.
2. A. Efrat and S. Har-Peled, *Guarding galleries and terrains*, IPL, 100 (2006), 238-245. Running time and the approx. ratio: (i) For simple polygons, $O(nc_{opt}^2 \log^4 n)$ expected time and $O(\log c_{opt})$, where c_{opt} is the size of the optimal solution. (ii) For polygons with h holes, $O(nhc_{opt}^3 polylog\ n)$ expected time and $O(\log n \log(c_{opt} \log n))$.
3. A. Deshpande, T. Kim, E. D. Demaine1 and S. E. Sarma, *A pseudopolynomial time $O(logn)$-approximation algorithm for art gallery problems*, Proceedings of the 10th WADS LNCS, Springer, no. 4619, pp. 163-174, 2007. Running time: Polynomial in n, the number of walls and the spread, where the spread can be exponential. Approx. ratio: $O(\log c_{opt})$.

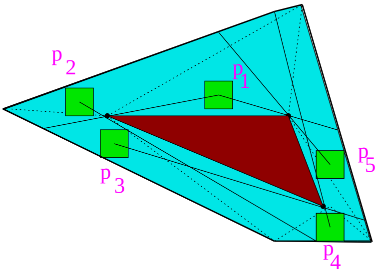
Optimal exploration and the Art Gallery Problem

- Suppose an optimal exploration algorithm for a point robot has computed visibility polygons from points p_1, p_2,...., p_k.
- We know that (i) $\bigcup_{i=1}^{k} V(P, p_i) = P$, (ii) $p_i \in V(P, p_j)$ for some $j < i$ and (iii) k is minimum. So, P can be guarded by placing stationary guards at p_1, p_2,...., p_k.
- The exploration problem for a point robot is the Art Gallery problem for stationary guards with additional constraint (ii).
- Our exploration algorithm for a point robot is an approximation algorithm for this variation of the Art Gallery problem.
- The exploration path of the robot in P is a watchman route or an autonomous inspection path.

Open Problem

Can one prove that the exploration problem, like the Art Gallery problem, is NP-hard?

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.

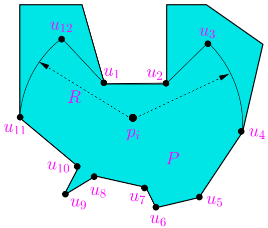
Open problem

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

Exploring an unknown polygon: Bounded visibility

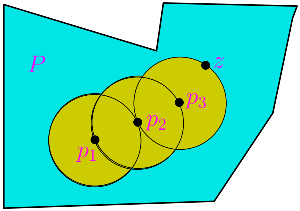
Computer vision range sensors or algorithms, such as stereo or structured light range finder, can reliably compute the 3D scene locations only up to a depth R. The reliability of depth estimates is inversely related to the distance from the camera. Thus, the range measurements from a vision sensor for 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 $u_1, u_2, \ldots, u_{12}$.

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, 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.

Concluding remarks

In this talk, we have reviewed a few algorithms for robot path planning in the plane which are based on visibility computations and have suggested a few open problems.