RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK I

Elmar Langetepe

Online Motion Planning

MA INF 1314

Summersemester 2016
Manuscript: Elmar Langetepe

Chapter 4

Exploration in polygons

We would like to consider the exploration task for polygons by an agent equipped with a vision system. The results can be applied to the framework of the preceding section. We are searching for a short path that *sees* all points in the polygon at least once. For a simple polygon the overall shortest such paths can be computed in polynomial time, if the polygon is given. There are also online algorithms that explores an a priori unknown simple polygon by a constant competitive strategy in comparison to the shortest offline path. For polygons with obstacles (holes) no such algorithms exist.

4.1 Simple polygons

A simple polygons is enclosed by a simple polygonal chain without self intersections. In the competitive sense we compare online exploration strategies with offline strategies.

The problem of computing the shortest round trip that sees all points in the polygon was introduced by 1986 by Chin and Ntafos as the Shortest Watchman problem; see [CN86]. Since then many authors have considered the Shortest Watchman Route (SWR) problem, sone of which have been erroneous. Other have been improved in the running time. Currently, it is meant to be common sense that the following result gives the best algorithm.

Theorem 4.1 *(Dror, Efrat, Lubiw, Mitchell, 2003)*
For a simple polygon with n vertices and a start point s, there is an algorithm that computes the Shortest Watchman Route in time $O(n^3 \log n)$. *[DELM03]*

First, we consider simple polygons and within this class of polygons special subclasses; see Figure **??**. Polygons of these classes allow efficient computations.

Definition 4.2 A simple polygon P is denoted as **monotone**, if there exists a line ℓ, such that for any line l' orthogonal to l the intersection $P \cap l'$ is path-connected. This means that the intersection $P \cap l'$ is a single segment, a point or empty. If ℓ is in parallel to the Y-axis, the polygon P is denotde as **y–monotone**.
A simple polygon P is denoted as **rectinlinear**, if any inner angle is either of 90° or of 270°.

The most simple case for the computation of a SWR is given for monotone and rectilinear polygons:

Theorem 4.3 *(Chin, Ntafos, 1986)*
For a rectilinear and monotone polygon, the SWR can be computed in $O(n)$ time. *[CN86, CN88]*

Exercise 23 *Present a linear time algorithm for the proof of the above Theorem.*

In general, for the computation of the SWR we can concentrate on discrete parts of the polygon. It suffices to visit the *essential cuts*, defined as follows. The invisible parts of the polygons lie behind reflex vertices, i.e., vertices with inner angle larger than π.

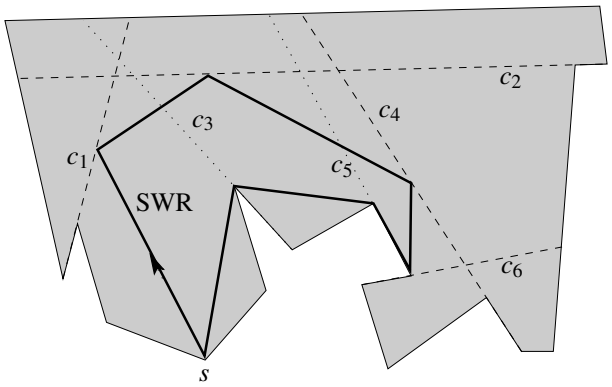


Figure 4.1: A polygon and necessary cuts (dotted), essential cuts (dashed) and the Shortest Watchman Route.

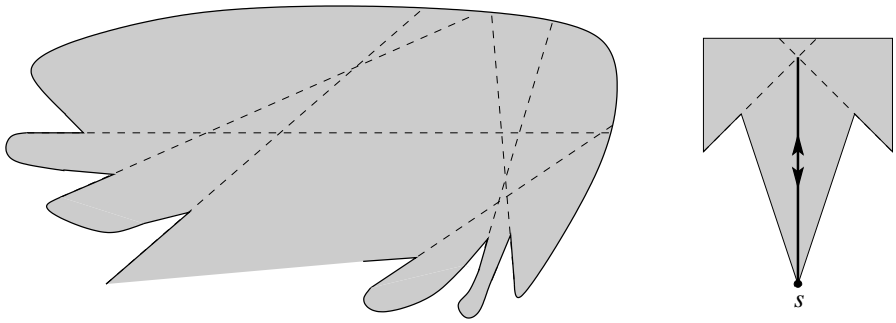


Figure 4.2: (i) A "corner" situation: Several cuts intersect and in a row and a single cut intersects more than one other cut. (ii) A polygon and its SWR.

Definition 4.4 Consider the extension of a an edge of a reflex vertex that points into the inner part of the polygon until it hits the boundary. Such segments are denoted as **cuts**. For the two cuts starting at a reflex vertex the cut emanating from the invisible edge (w.r.t. the start point) has to be crossed in order to see both edges. These cuts are called **necessary cuts**. For a necessaryl cut c_i let P_{c_i} denote the sub-part of the polygon P behind c_i w.r.t. the start point. The agent has to move inside P_{c_i}. A necessary cut c_i **dominates** a necessary cut c_j if $P_{c_i} \subset P_{c_j}$ holds. In this case any path from the start that visits P_{c_i} visits the larger polygon P_{c_j} first. A necessary cut c_i that is not dominated by any other necessary cut is denoted as an **essential** cut. It is clear that for the SWR it is sufficient to visit all essential cuts.

Necessary cuts that will be dominated will be explored on the path to the corresponding essential cut. Figure 4.1 shows an example with necessary and essential cuts. Here c_3 and c_5 are not essential, any path to c_4 will visit the cuts. More precisely, c_4 dominates c_3 and c_5. With the help of the cuts we can formulate some structural properties:

(i) The SWR and any other exploration tour has to visit all essential cuts. The set of essential cuts is the smallest set of cuts that has to be visited for seeing the whole polygon.

(ii) If the essential cuts do not intersect, they have to be visited in their order along the boundary. In this case from the SWR the cuts will be visited by specular reflection. The incoming angle for the visit of each cut is the same as the outgoing angle.

(iii) If some essential cuts intersect in a row, we call this a "Corner" situation. In this case it might happen that some cuts are just passed by the SWR and are not visited by specular reflection; see Figure 4.2. This makes the corner situation difficult.

For a polygon and a start point s we can order the cuts by the order they appear along the boundary, independent from the position of the corresponding reflex vertex; see Figure 4.1. In the corner situation

the SWR need not visit the essential cuts in this order; see Figure 4.3.

Interestingly, the corresponding polygons P_{c_i} are still visited in the order of the corresponding cuts. In Figure 4.3 we have the visiting order $P_1, P_2, P_3, P_4, P_5, P_6$. This is meant as follows.

Although, we first enter P_3 the SWR actually visits P_1 and P_2 at a single point first. By chance we are also in P_3 at this point, visit P_3 immediately and the order is maintained in this sense. Pre-visits do not count. This means that the task is: Computed the shortest tour that visits the polygons P_{c_i} by the order along the boundary.

We will pick up this idea later on. First we consider the simple case of a rectilinear polygon. In a rectilinear polygon everything is less complicated. We do not have complicated corner situations. Essential cuts have successive intersections for max three orthogonal cuts; see Figure 4.3. We conclude.

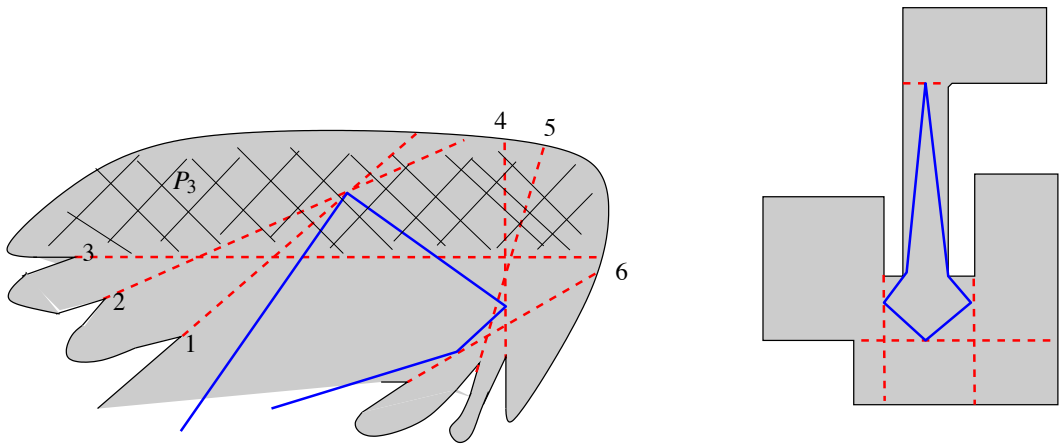
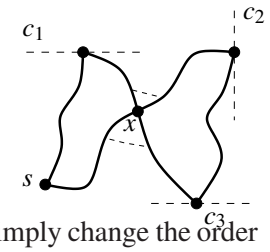


Figure 4.3: In a corner situation the SWR visits the polygons P_{c_i} (here P_i) by the order of the corresponding essential cuts along the boundary. In rectilinear polygon essential cuts will never be passed.

Lemma 4.5 *For a rectlinear polygons the SWR visits the essential cuts by the order along the boundary.*

Proof. For the rectinlinear case a corner situation can occur with maximal three cuts, where the first and the last one run run in parallel and do not intersect. Moving into the corresponding polygons P_{c_i} gives a detour. It is needless to pass a cut in order to reach another cut. Therefore all cuts will be visited one after the other.



Assume that the visits do not follow the order along the boundary. In this case the SWR R will have an intersection somewhere; see the Figure. We can simply change the order locally in order to obtain R' that runs from s c_1, x, c_2, c_3, x to s. This is also a tour that has the same property. In the vicinity of the intersection x we can even locally (and globally) improve the tour by some shortcuts. □

Lemma 4.5 gives the key-idea for the computation of the SWR:

Theorem 4.6 *(Chin, Ntafos, 1986)*
The Shortest Watchman Route in a simple, rectilinear polygon can be computed in $O(n)$ time. [CN86, CN88]

Proof. Algorithm 4.1 computes the SWR in a rectilinear polygon, Figure 4.4 shows an example.

The essential cuts can be commputed in $O(n)$ time (exercise left to the reader). It has to be shown that P'', has no more than $O(n)$ edges or triangles. All other running times stem from standard offline algorithms for polygons. We consider dual graph, T^*, of the triangulation. Successively visiting the cuts along the corresponding triangles is simply Depth-First runs through T^*. Any edge of T^* is visited exactly twice. This means that also any triangle occur only twice in P'', the number of triangles and edges in P'' is in $O(n)$. □

Exercise 24 *Show that the essential cuts in a rectilinear polygon of n vertices can be computed in $O(n)$ time.*

Algorithm 4.1 Shortest Watchman Route for rectilinear polygons

- Compute the essential cuts $c_1, \ldots, c_k$ and order them along the boundary from s. $O(n)$
- Cut of the corresponding sub-polygons P_{c_i} behind the cuts. This gives a polygon P'. Some of the essential cuts get smaller. $O(n)$
- Compute a triangulation for P'. $O(n)$
- Compute a chain of triangles P'' by the "Roll–Out" of P': $O(n)$:
  - Let $P^{(1)}$ be the relevant triangles of P' along the path in the dual graph of the triangulation, T^* , from s to c_1.
  - For any essential cut c_i, $i = 2, \ldots, k$: Extend $P^{(i-1)}$ to $P^{(i)}$ by the chain of the relevant triangles along the boundary of P' on the path from c_{i-1} to c_i and reflected at the c_{i-1}.
  - Extend $P^{(k)}$ to P'' as above by the relevant triangles on the path from c_k to s and by reflection on c_k. There will be a copy s' of s.
- P'' is a sequence of triangles. Compute the shortest path π from s to s' in P''. $O(n)$
- The SWR can be build by mirroring back the line segments of the path at the cuts c_i.

Algorithm 4.1 can be applied to any polygon in the same way, if any essential cut of the polygon intersect with exactly one other essential cut. In this case Lemma 4.5 holds. In general polygons this will not be the case. Many essential cuts can intersect in a row with multiple intersections of a single cut with others. We call such situations a "corner" situation. In a corner, the order of the visits of the cuts is non-trivial.

First, we would like to argue, that the above algorithm can be easily made depth-restricted. For this we only have to restrict the set of essential cuts. An essential cut blocks the visibility of points closely behind the reflex vertex of the cut. We consider a non-visible point that has the closest distance to the start s. In principle this point is arbitrary close to the reflex vertex. So the distance to the reflex vertex gives the distance to the cut. In Figure 4.5(i) the rightmost essential cut has distance $l > d$.

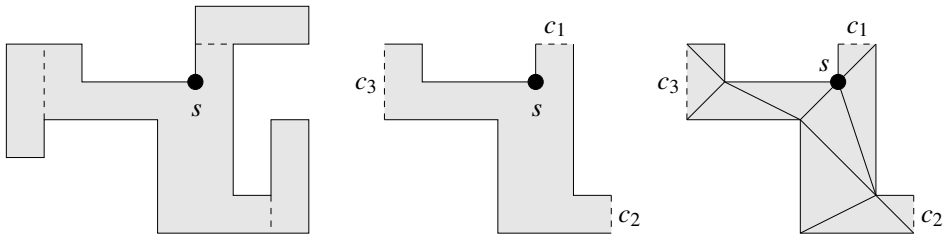
We would like to see all points in P with distance less than or equal to d from s. Let $P(d)$ denote this part of P. Obviously, it is sufficient to visit all essential cuts that has a distance $\leq d$. $> d$ to the start s; see Figure 4.5(ii). We apply the same algorithm.

For simple, rectilinear polygons we conclude: $\mathrm{Expl}_{\mathrm{OFF}}(d) = \mathrm{Expl}_{\mathrm{opt}}(d)$. This means that for the offline case we have $\beta = 1$ and $C_\beta = 1$ for the exploration of $P(d)$ and the application of Theorem 3.24 gives an 8-approximation of the optimal search ratio. Suchpfades.

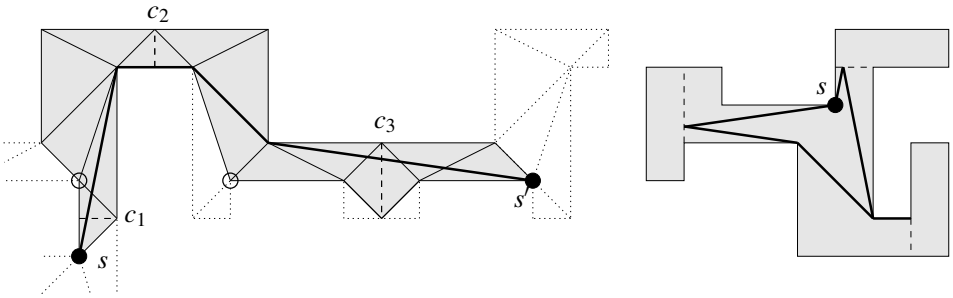
Algorithm 4.2 Online exploration of a rectlilinear polygon

while Polygon is not fully explored **do**
  Consider the next reflex vertex along the boundary in cw order.
  Move orthogonally to the corresponding cut.
end while

In the online version of the problem, the poygon is a priori not known. Nevertheless, we can design an efficient online algorithm. There are no corner situations and we can visit the cuts of the reflex vertices by the Greedy-Algorithm 4.2; see also Figure 4.6. Starting from s at the boundary we successively expand the visible part of the boundary and always approach the next reflex vertex by a move orthogonal to its cut. This gives an L_1-optimal exploration path. We have the following result:

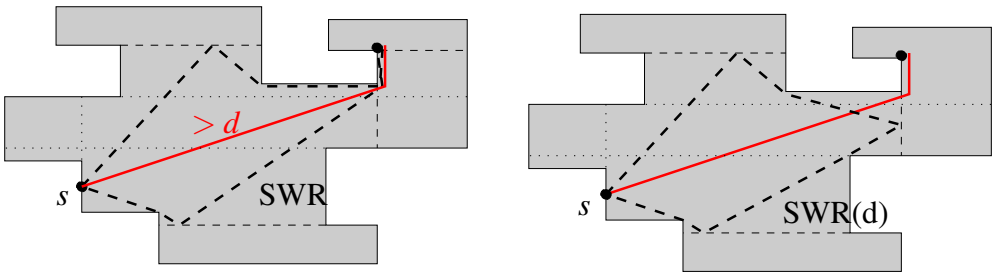


(i) Wesentliche Cuts (ii) ”Au”seres abschneiden (iii) Triangulation



(iv) Roll-Out und k”urzester Weg (v) Shortest Watchman Route

Figure 4.4: Computing the SWR in a rectinlinear polygon.



(i) (ii)

Figure 4.5: Computation of the SWR for all points with distance $\leq d$ from s in a rectilinear polygon. It is sufficient to ignore all cuts of distance $> d$.

Theorem 4.7 *(Deng, Kameda und Papadimitriou, 1991)*
A simple, rectilinear polygon can be explored online optimally w.r.t. the L_1-metric and with a competitive ratio of $\sqrt{2}$ w.r.t. the L_2-metric [1]. *[DKP98]*

Proof. We give a sketch of the proof. The Greedy-exploration approach give an optimal L_1-path, since the algorithm successively creates locally optimal L_1-paths. In the first step the first cut will be visited orthogonally by an optimal L_1-paths. Assume that we are already along an optimal L_1-path and have visited a set of cuts in this fashion. The next cut is again visited orthogonally on the shortest L_1-path. By induction the agent moves along an overall shortest L_1-path for visiting the necessary cuts.

We still have to move back. For this we simply assume that at the start point s there is an *artificial* necessary cut. Also this last cut will be visited by an optimal L_1-path, which gives an overall optimal L_1 round trip.

For the comparison to the optimal L_2-SWR, we use the following sketch. Also the L_2-SWR visits the

[1] For the L_1-metric or Manhatten-metric the distance between two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ is defined by $d(p,q) := |p_x - q_x| + |p_y - q_y|$; in the L_2- or Euclidean metric we have $d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$.

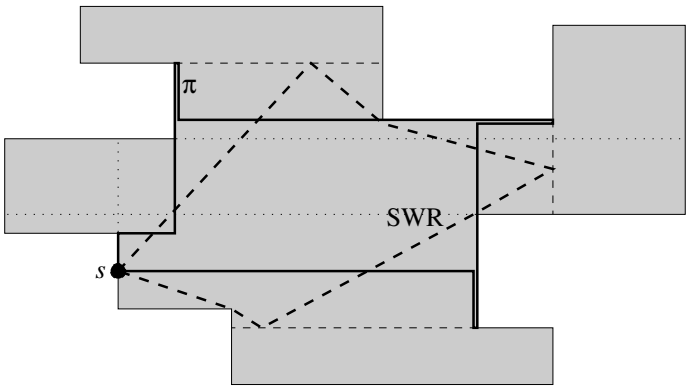


Figure 4.6: Path of the online heuristic and the SWR w.r.t. the L_2-metric in a rectilinear polygon.

essential cuts in the order along the boundary. We shift the L_1-path to the outer boundary such that the L_2 SWR path is included and the L_1-path still has the same length. For any two point of a segment of the L_2-SWR there is an optimal L_1-path which can be considered to consist of two segments, we only have to check detours of triangles; see Figure 4.8.

Thus, we consider a single triangle and by scaling we can argue that we have to consider the maximum of the $f(x,y) = x+y$ for $x^2+y^2 = 1$. This means that $f(y) = y+\sqrt{1-y^2}$ has to be maximized. We have $f'(y) = 1-\frac{y}{\sqrt{1-y^2}}$ and the f' gets 0 for $y_{\max} = \frac{1}{\sqrt{2}}$. This is a maximum of f and we have $x_{\max} = \frac{1}{\sqrt{2}}$ and $f(x_{\max}, y_{\max}) = \sqrt{2}$. □

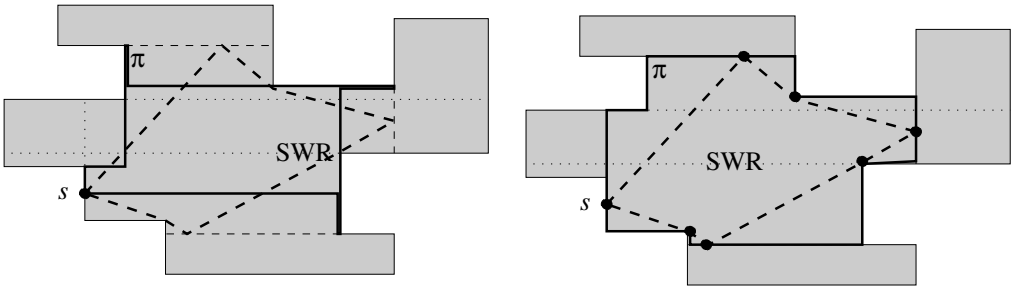


Figure 4.7: *Shifting* an L_1-optimal path, such that the L_2-SWR is inside. The analysis of the detour for triangles is sufficient.

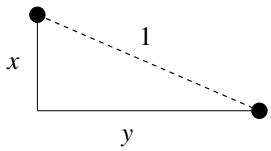


Figure 4.8: The worst-case detour in a triangle is $\sqrt{2}$.

Bibliography

[Ad80] H. Abelson and A. A. diSessa. *Turtle Geometry*. MIT Press, Cambridge, 1980.

[AFM00] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, 17:25–50, 2000.

[AG03] Steve Alpern and Shmuel Gal. *The Theory of Search Games and Rendezvous*. Kluwer Academic Publications, 2003.

[AKS02] Susanne Albers, Klaus Kursawe, and Sven Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32:123–143, 2002.

[BRS94] Margrit Betke, Ronald L. Rivest, and Mona Singh. Piecemeal learning of an unknown environment. Technical Report A.I. Memo No. 1474, Massachusetts Institute of Technology, March 1994.

[BSMM00] Ilja N. Bronstein, Konstantin A. Semendjajew, Gerhard Musiol, and Heiner Mühlig. *Taschenbuch der Mathematik*. Verlag Harry Deutsch, Frankfurt am Main, 5th edition, 2000.

[BYCR93] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Inform. Comput.*, 106:234–252, 1993.

[CN86] W. Chin and S. Ntafos. Optimum watchman routes. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 24–33, 1986.

[CN88] W. Chin and S. Ntafos. Optimum watchman routes. *Inform. Process. Lett.*, 28:39–44, 1988.

[DELM03] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell. Touring a sequence of polygons. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 473–482, 2003.

[DHN97] G. Das, P. Heffernan, and G. Narasimhan. LR-visibility in polygons. *Comput. Geom. Theory Appl.*, 7:37–57, 1997.

[DJMW91] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *Transactions on Robotics and Automation*, 7:859–865, 1991.

[DKK01] Christian A. Duncan, Stephen G. Kobourov, and V. S. Anil Kumar. Optimal constrained graph exploration. In *Proc. 12th ACM-SIAM Symp. Discr. Algo.*, pages 307–314, 2001.

[DKK06] Christian A. Duncan, Stephen G. Kobourov, and V. S. Anil Kumar. Optimal constrained graph exploration. *ACM Trans. Algor.*, 2:380–402, 2006.

[DKP98] Xiaotie Deng, Tiko Kameda, and Christos Papadimitriou. How to learn an unknown environment I: The rectilinear case. *J. ACM*, 45(2):215–245, 1998.

[EFK+06] Andrea Eubeler, Rudolf Fleischer, Tom Kamphans, Rolf Klein, Elmar Langetepe, and Gerhard Trippen. Competitive online searching for a ray in the plane. In Sándor Fekete, Rudolf Fleischer, Rolf Klein, and Alejandro López-Ortiz, editors, *Robot Navigation*, number 06421 in Dagstuhl Seminar Proceedings, 2006.

[FKK+04] Rudolf Fleischer, Tom Kamphans, Rolf Klein, Elmar Langetepe, and Gerhard Trippen. Competitive online approximation of the optimal search ratio. In *Proc. 12th Annu. European Sympos. Algorithms*, volume 3221 of *Lecture Notes Comput. Sci.*, pages 335–346. Springer-Verlag, 2004.

[Gal80] Shmuel Gal. *Search Games*, volume 149 of *Mathematics in Science and Engeneering*. Academic Press, New York, 1980.

[GKP98] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley, 1998.

[GR03] Yoav Gabriely and Elon Rimon. Competitive on-line coverage of grid environments by a mobile robot. *Comput. Geom. Theory Appl.*, 24:197–224, 2003.

[HIKL99] Christoph Hipke, Christian Icking, Rolf Klein, and Elmar Langetepe. How to find a point on a line within a fixed distance. *Discrete Appl. Math.*, 93:67–73, 1999.

[IKKL00a] Christian Icking, Thomas Kamphans, Rolf Klein, and Elmar Langetepe. Exploring an unknown cellular environment. In *Abstracts 16th European Workshop Comput. Geom.*, pages 140–143. Ben-Gurion University of the Negev, 2000.

[IKKL00b] Christian Icking, Thomas Kamphans, Rolf Klein, and Elmar Langetepe. Exploring an unknown cellular environment. Unpublished Manuscript, FernUniversität Hagen, 2000.

[IKKL05] Christian Icking, Tom Kamphans, Rolf Klein, and Elmar Langetepe. Exploring simple grid polygons. In *11th Internat. Comput. Combin. Conf.*, volume 3595 of *Lecture Notes Comput. Sci.*, pages 524–533. Springer, 2005.

[IKL97] Christian Icking, Rolf Klein, and Elmar Langetepe. Searching for the kernel of a polygon: A competitive strategy using self-approaching curves. Technical Report 211, Department of Computer Science, FernUniversität Hagen, Germany, 1997.

[IKL99] Christian Icking, Rolf Klein, and Elmar Langetepe. An optimal competitive strategy for walking in streets. In *Proc. 16th Sympos. Theoret. Aspects Comput. Sci.*, volume 1563 of *Lecture Notes Comput. Sci.*, pages 110–120. Springer-Verlag, 1999.

[IKL+04] Christian Icking, Rolf Klein, Elmar Langetepe, Sven Schuierer, and Ines Semrau. An optimal competitive strategy for walking in streets. *SIAM J. Comput.*, 33:462–486, 2004.

[IPS82] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM J. Comput.*, 11:676–686, 1982.

[KL03] Tom Kamphans and Elmar Langetepe. The Pledge algorithm reconsidered under errors in sensors and motion. In *Proc. of the 1th Workshop on Approximation and Online Algorithms*, volume 2909 of *Lecture Notes Comput. Sci.*, pages 165–178. Springer, 2003.

[Kle91] Rolf Klein. Walking an unknown street with bounded detour. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 304–313, 1991.

[Kle97] Rolf Klein. *Algorithmische Geometrie*. Addison-Wesley, Bonn, 1997.

[KPY96] Elias Koutsoupias, Christos H. Papadimitriou, and Mihalis Yannakakis. Searching a fixed graph. In *Proc. 23th Internat. Colloq. Automata Lang. Program.*, volume 1099 of *Lecture Notes Comput. Sci.*, pages 280–289. Springer, 1996.

[Lan00] Elmar Langetepe. *Design and Analysis of Strategies for Autonomous Systems in Motion Planning*. PhD thesis, Department of Computer Science, FernUniversität Hagen, 2000.

[LB99] Sharon Laubach and Joel Burdick. RoverBug: Long range navigation for mars rovers. In Peter Corke and James Trevelyan, editors, *Proc. 6th Int. Symp. Experimental Robotics*, volume 250 of *Lecture Notes in Control and Information Sciences*, pages 339–348. Springer, 1999.

[Lee61] C. Y. Lee. An algorithm for path connections and its application. *IRE Trans. on Electronic Computers*, EC-10:346–365, 1961.

[LOS96] Alejandro López-Ortiz and Sven Schuierer. Walking streets faster. In *Proc. 5th Scand. Workshop Algorithm Theory*, volume 1097 of *Lecture Notes Comput. Sci.*, pages 345–356. Springer-Verlag, 1996.

[LS87] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[Sch01] S. Schuierer. Lower bounds in on-line geometric searching. *Comput. Geom. Theory Appl.*, 18:37–53, 2001.

[Sha52] Claude E. Shannon. Presentation of a maze solving machine. In H. von Foerster, M. Mead, and H. L. Teuber, editors, *Cybernetics: Circular, Causal and Feedback Mechanisms in Biological and Social Systems, Transactions Eighth Conference, 1951*, pages 169–181, New York, 1952. Josiah Macy Jr. Foundation. Reprint in [Sha93].

[Sha93] Claude E. Shannon. Presentation of a maze solving machine. In Neil J. A. Sloane and Aaron D. Wyner, editors, *Claude Shannon: Collected Papers*, volume PC-03319. IEEE Press, 1993.

[SM92] A. Sankaranarayanan and I. Masuda. A new algorithm for robot curvefollowing amidst unknown obstacles, and a generalization of maze-searching. In *Proc. 1992 IEEE Internat. Conf. on Robotics and Automation*, pages 2487–2494, 1992.

[SS99] Sven Schuierer and Ines Semrau. An optimal strategy for searching in unknown streets. In *Proc. 16th Sympos. Theoret. Aspects Comput. Sci.*, volume 1563 of *Lecture Notes Comput. Sci.*, pages 121–131. Springer-Verlag, 1999.

[Sut69] Ivan E. Sutherland. A method for solving arbitrary wall mazes by computer. *IEEE Trans. on Computers*, 18(12):1092–1097, 1969.

[SV90a] A. Sankaranarayanan and M. Vidyasagar. A new path planning algorithm for a point object amidst unknown obstacles in a plane. In *Proc. 1990 IEEE Internat. Conf. on Robotics and Automation*, pages 1930–1936, 1990.

[SV90b] A. Sankaranarayanan and M. Vidyasagar. Path planning for moving a point object amidst unknown obstacles in a plane: A new algorithm and a general theory for algorithm developments. In *Proceedings of 1990 IEEE Conf. on Decision and Control*, pages 1111–1119, 1990.

[SV91] A. Sankaranarayanan and M. Vidyasagar. Path planning for moving a point object amidst unknown obstacles in a plane: The universal lower bound on the worst case path lengths and a classification of algorithms. In *Proc. 1991 IEEE Internat. Conf. on Robotics and Automation*, pages 1734–1741, 1991.

[THL98] L. H. Tseng, P. Heffernan, and D. T. Lee. Two-guard walkability of simple polygons. *Internat. J. Comput. Geom. Appl.*, 8(1):85–116, 1998.

[Wal86] Wolfgang Walter. *Gewöhnliche Differentialgleichungen*. Springer, 1986.

[Web07] Maximilian Weber. Online suche auf beschränkten sternen. Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn, 2007.

Index

T

U

V

W

Y