A New Lower Bound for Kernel Searching*

Peter Anderson   Alejandro López-Ortiz †

Abstract

We consider the problem of on-line searching for the kernel of a unknown star-shaped polygon. In this motion planning problem, the robot starts from a point s inside a simple star-shaped polygon P, and aims to reach the kernel of P. The robot has no knowledge of P (apart from the fact that is a star-shaped polygon) but it is equipped with an on-board vision system that allows it to see its surrounding space. We prove that any strategy for this purpose in the worst case must traverse at least 1.55 times the shortest distance from s to the kernel of P. This improves over the best previously known lower bound of 1.44 by López-Ortiz and Schuierer.

1 Introduction

In recent years on-line searching has been an active area of research in Computer Science (e.g. [1, 2, 3]). In its full generality, an on-line search problem consists of an agent searching for a target area in an unknown terrain. In the worst case a search by a robot in a general domain can be arbitrarily inefficient as compared to the shortest path from the initial position to the target. However, as it is to be expected, strategies can be improved depending on the type of terrain and the searching capabilities of the robot.

In this work we assume that the robot is equipped with an on-board vision system that allows it to see its local environment. Since the robot has to make decisions about the search based only on the part of its environment that it has seen before, the search of the robot can be viewed as an *on-line* problem. As such, the performance of an on-line search strategy can be measured by comparing the distance traveled by the robot with the length of the shortest path from the starting point s to the target area A. The worst case ratio of the distance traveled by the robot to the optimal distance from s to A is called the *competitive ratio* of the search strategy.

Icking and Klein studied the problem of on-line searching for the *kernel* of a star-shaped polygon. In this case, the competitive ratio is given by the ratio of the length traversed by the robot from the starting point s to the closest kernel point p to the distance from s to p. They present a strategy with a competitive ratio of ~ 5.331 [6], which was later shown to be exactly $\pi + 1$ competitive [9]. A strategy with a competitive ratio bounded by $1 + 2\sqrt{2} \sim 3.829$ was later given by J-H. Lee et al. [10] and recently improved to ~ 3.1226 by L. Palios [13].

Icking and Klein also pointed out that the natural $\sqrt{2}$ lower bound applies to this problem. Later López-Ortiz and Schuierer improved this bound to ~ 1.44.

In this work we show a lower bound of ~ 1.55 for any strategy that finds the kernel of a star-shaped polygon.

*This research is partially supported by an NSERC Research Grant.

†Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada, E3B 4A1, email: alopezo@unb.ca

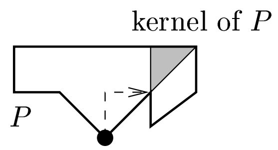

Figure 1: Lower bound configuration for walking into the kernel.

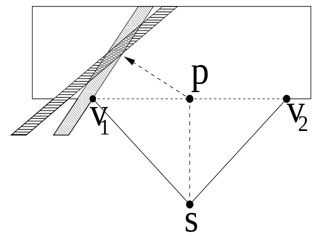

Figure 2: Polygon with two beams.

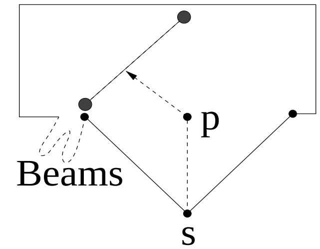

Figure 3: Lower bound configuration.

2 Definitions

We say two points p and p' in a polygon P are visible to each other if the line segment $\overline{pp'}$ is contained in P. If A and B are two sets, then A is weakly visible from B if every point in A is visible from some point in B. The visibility polygon of p is the subset of points in P that are visible to p; it is denoted by $V_P(p)$. We assume that the robot has access to its local visiblity polygon by a range sensing device, e.g. a ladar. Now we can define a star-shaped polygon.

Definition 1 ([14]) *A simple polygon P is a* star-shaped polygon *if there exists a point p in P such that $V_P(p) = P$. The set of all points p inside P with $V_P(p) = P$ is the* kernel *of P.*

If the robot does not start in the kernel of P, then there are regions in P that cannot be seen by it. The connected components of $P \setminus V_P(p)$ are called pockets. The boundary of a pocket is made of some polygon edges and one line segment that does not belong to the boundary of P—which is called a window of $V_P(p)$. Note that a window intersects the boundary of P only in its end points. More generally, a line segment that intersects the boundary of P only in its end points is called a chord.

A pocket is said to be a left pocket if it lies locally to the left of the pocket ray that contains its window. A pocket edge is said to be a left pocket edge if it defines a left pocket. An extended pocket edge is a left extended pocket edge if its first line segment is collinear with a left pocket edge. Right pocket, right pocket edge, and right extended pocket edge are defined analogously.

Since a point in the kernel of P sees all the points in P, in particular p, a pocket of $V_P(p)$ does not intersect the kernel of P which implies the following observation.

3 Walking into the Kernel—a Lower Bound

In this section we consider the problem of on-line searching and walking into the kernel of a star-shaped polygon [6, 10, 9, 11]. We present a lower bound of ~ 1.55 on the competitive ratio of any strategy to walk into the kernel.

Figure 1 shows a lower bound of $\sqrt{2}$. Any on-line strategy with a competitive ratio of $\leq \sqrt{2}$ has to follow the dashed path [6]. In the following we show that any strategy to search for the kernel of a star-shaped polygon has a competitive ratio that is significantly larger than $\sqrt{2}$.

Definition 2 *The visibility region of a subset B of a polygon is the set of all points in the polygon which see all points in B.*

Definition 3 *Given the current position of the robot p and a pocket B of $V_P(p)$, the* beam *of the pocket is the visibility region of B.*

Notice that if the pocket is a trapezoid, the visibility region resembles a search light beam (see Figure 2).

Observation 1 *The kernel lies in the intersection of all beams.*

We now prove a lower bound by using an interesting numerical analysis technique. First, we compute a discrete approximation to an optimal search path, and then, by bounding the numerical error of the approximation through a formal mathematical proof, we obtain a rigorous lower bound on the optimal path.

Theorem 1 *Walking into the kernel of a polygon is at least* 1.55*-competitive.*

Proof. Consider the polygon of Figure 2. Notice that the robot reaches the line segment $\overline{v_1v_2}$ before it reaches the kernel. The robot reaches $\overline{v_1v_2}$ at a point p. From p it is not yet clear where the kernel is located. In fact, depending upon the specific angle and location of the pockets, the beams might specify a small kernel located anywhere in the visibility polygon region of s which is above $\overline{v_1v_2}$.

We now use an adversary argument. After the robot reaches p the adversary closes one side, and leaves two beams on the other. One beam selects a ray along which the kernel will be located and the second one determines if the kernel is on one end of the ray or on the other end. This is illustrated by the large dots in Figure 3. This can be achieved by locating one beam A along the line joining the two candidate regions, and a second beam, B, nearly parallel and to the right of A. The intersection of both beams defines the kernel of the polygon.

The angle determined by the beam and the line $\overline{v_1v_2}$ can range anywhere between 0 and $\pi/2$. The optimal strategy follows a path Γ. We introduce a 1000×30 grid in the polar coordinates interval $[0, 2] \times [0, \pi/2]$ and approximate the path Γ with a path $\widehat{\Gamma}$ on the grid. Then a program computes first a crude upper bound on the optimal solution and uses this bound to bootstrap an exhaustive search with pruning over the space of all paths on the grid. This produces an improved approximation which is iteratively further enhanced by introducing a refined grid in the vicinity of this path, eventually resulting on a grid of 1000 points on an interval of length 0.01 units. This program, written in C using double precision arithmetic, obtains a path on the grid of length 1.556.

There are two aspects of the numerical computation that need to be bound. The error of the approximation even if infinite precision arithmetic was used and the error of computing the approximation itself. In the field of Numerical Analysis, the first type of error is known as approximation error while the second type of error is termed round-off error [5].

To bound the approximation error observe that the optimal path crosses each ray between two points of the grid. Therefore the exhaustive search algorithm considered paths made of any of the four possible combinations connecting the two points from one ray with the two points of the next. Let γ_i be the length of the optimum path from ray i to ray $i+1$ and let $\hat{\gamma}_i$ be the length of the approximation path between the same two rays. Given a point on a ray let $1+\delta$ be the position of the immediate neighbour in the grid for that ray. Using polar coordinates and elementary trigonometrics one can show that

$$\frac{\hat{\gamma}}{\gamma} \leq 1 + \delta.$$

Therefore

$$|\Gamma| = \sum_i \gamma_i \leq \sum_i (1+\delta)\hat{\gamma}_i = (1+\delta)|\widehat{\Gamma}|,$$

which implies that the approximation error is at most $1+\delta$.

The roundoff error is the effect of using finite precision arithmetic for computations on the real numbers. In this case, the length of $\widehat{\Gamma}$ was computed on a Pentium III system using Intel built-in numeric functions. On a single computation these functions produce results with 53 bits of precision [4, 12] for elementary numeric functions used such as $\sqrt{}$, sin() and cos().

In general, this does not suffice to bound the roundoff error, as it may accumulate on each succesive iteration. However, in the specific case of the computation of the entire path length, each step length is computed independently and then added over the thirty rays composing the grid. On each individual step arithmetics show that the error is no larger than $(1+\epsilon)^4$ where ϵ is the precision of the built-in functions. Thus the total accumulated error is no larger than $30(1+\epsilon)^4$. Substituting for the proper value of ϵ and using the approximation found by the program we obtain that the optimal path is at least 1.55 the optimal shortest path between the starting position of the robot and the kernel. □

4 Conclusions

We show that no strategy which walks into the kernel of a star-shaped polygon can do better than 1.55 which improves on the best previously known lower bound of 1.44.

A gap of 1.6 remains between the best upper and lower bound for this problem.

References

[1] R. Baeza-Yates, J. Culberson and G. Rawlins. "Searching in the plane", *Information and Computation*, Vol. **106**, (1993), pp. 234-252.

[2] P. Berman, A. Blum, A. Fiat, H. Karloff, A. Rosén and M. Sacks. "Randomized robot navigation algorithms", *Proc. of 7th ACM-SIAM Symp. on Disc. Algs.*, (1996).

[3] A. Blum, P. Raghavan and B. Schieber. "Navigating in unfamiliar geometric terrain", *Proc. of 23rd ACM Symposium on Theory of Computing*, (1991), pp. 494-504.

[4] J. Harrison, T. Kubaska, S. Story, P.T.P. Tang. "The Computation of Transcendental Functions on the IA-64 Architecture." *Intel Technology Journal*, Q4 1999.

[5] M.T. Heath. *Scientific Computing: An Introductory Survey*, McGraw-Hill, 1997.

[6] Ch. Icking and R. Klein. "Searching for the kernel of a polygon. A competitive strategy", *Proc. 11th ACM Symposium on Computational Geometry*, (1995).

[7] R. Klein. "Walking an unknown street with bounded detour", *Computational Geometry: Theory and Applications*, Vol. **1**, (1992), pp. 325-351.

[8] D.T. Lee and F.P. Preparata. "An optimal algorithm for finding the kernel of a polygon", *Journal of the ACM*, Vol. **26**, (1979), pp. 415-421.

[9] J-H. Lee, K-Y. Chwa. "Tight Analysis of a Self-Approaching Strategy for the Online Kernel-Search Problem", *Inf. Proc. Let.*, Vol. 69, No. 1, 1999, pp.39-45.

[10] J-H. Lee, C-S. Shin, J.H. Kim and S.Y. Shin. "New Competitive Strategies for Searching in Unknown Star-Shaped Polygons", *Proc. 13th ACM Symp. on Comp. Geom.*, (1997).

[11] A. López-Ortiz and S. Schuierer, "Position-independent Near Optimal Searching and On-line Recognition in Star Polygons", *Proc. 5th WADS* , (1997), LNCS pp. 284-296.

[12] T. Lynch, M. Schulte. "A High Radix On-line Arithmetic for Credible and Accurate Computing". *Journal of Universal Computer Science.* Vol 1, No 7, 1995.

[13] L. Palios, "A New Competitive Strategy for Reaching the Kernel of an Unknown Polygon", To appear, *Proc. 7th SWAT*, 2000.

[14] F. P. Preparata, M. I. Shamos. *Computational Geometry,* Springer-Verlag, 1985.