Optimally Routing a Tracker to Maximize the Total Time a Mobile Evader is in View*

Gui Citovsky † Kan Huang † Joseph S. B. Mitchell †

Abstract

In this paper we consider a type of surveillance problem, which has been widely studied. Unlike [1] [2], which only maximize the time an evader is seen until it is first out of a tracker's vision, we try to maximize the total time an evader is seen as it moves on a known trajectory to its destination.

1 Introduction

Given the trajectory of an evader in polygon P and the starting location of a tracker, we compute a trajectory for the tracker that maximizes the amount of time the evader is seen.
We assume the following:

1. P is a rectilinear, simple n-gon
2. Rectangle vision: points $p, q \in P$ see each other iff $\square pq$, the axis-aligned bounding box of p, q, is a subset of P
3. The evader's trajectory is a shortest path within P
4. All trajectories are axis-aligned
5. The evader travels at constant speed and the tracker can travel no faster than the evader

Let $VP(p)$ denote the visibility polygon of point $p \in P$. Let ev be the evader, tr be the tracker, τ_e and τ_t be the trajectories of the evader and tracker respectively.

*Research partially supported by NSF (CCF-1018388), the US-Israel Binational Science Foundation (grant 2010074), and Sandia National Labs.

†Stony Brook University, Stony Brook, NY 11794

By extending the edges incident to reflex vertices, we partition P into basic (rectangular) regions [3]. Note that for any particular basic region c_i in this grid, each point inside this region has the same visibility polygon.
As ev moves along τ_e, we say that ev is *escaping from tr* if the distance from ev to the current position of tr is monotonically increasing.

2 Three-Stage Solution

In a simple polygon, the relationship between ev and tr has at most has three stages:

1. ev cannot see tr but is not escaping from tr
2. ev has seen tr at least once, and ev is not yet escaping from tr
3. ev is escaping from tr

2.1 Stage 3: Escaping

Let $[c_1, c_2, \ldots, c_r]$ be the ordered sequence of basic regions ev visits. The following algorithm has tr chasing the visibility polygons of ev.

Algorithm 1 Chasing Visibility Polygons

```
j = 1
while j ≤ r do
  while j ≤ r and tr ∈ VP(c_j) do
    j+ = 1
  if j ≤ r then
    send tr to VP(c_j) via a shortest path
  j+ = 1
```

Theorem 1 *With $O(n^2)$ preprocessing time, Algorithm 1 yields an optimal solution to Stage 3 in time $O(nr)$, r is the number of basic regions through which ev passes ($r = O(n)$).*

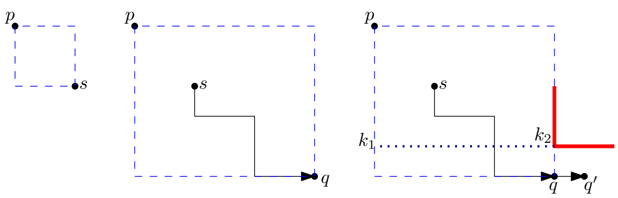

Figure 1: The moment ev leaves $VP(p)$.

Proof Refer to Figure 1. Initially, the tracker is at p and the evader is at s.

We assume that ev can see point p until it arrives at q, which means $\square pz$ is fully contained in the polygon for any point z on the path of the evader from s to q. When the evader moves a little further from p at q to q', q' cannot see p, which means the boundary of the polygon blocked the sight of q' as the rightmost part of Figure 1 shows; therefore a cut $\overline{k_1k_2}$ is created, which separates $VP(q')$ from p. It's easy to see that $\square pq$ is in $VP(z)$ for any point z on the evader's path from s to q. Both $\overline{k_1k_2}$ and p are in $\square pq$, thus if the tracker travels along $\overline{pk_1}$, he will see the evader at least until the evader leaves point q.

Since P is simple and τ_e is a shortest path, the tracker will always go to $VP(q')$ in order to have any hope of seeing the evader after the evader leaves point q. Observe that $\overline{pk_1}$ is the shortest path from p to $VP(q')$. The tracker sees the evader for as long as possible as he traverses $\overline{pk_1}$; therefore $\overline{pk_1}$ is the optimal path the tracker should follow.

The analysis is similar in the case that tr cannot see ev initially. ∎

This algorithm runs in time $O(nr)$. We compute $O(r)$ visibility polygons and shortest paths. Computing visibility polygons and shortest paths takes $O(n)$ time. Note that $r = O(n)$ because each extended edge can be charged to its corresponding vertex in P. Since τ_e is a shortest path, ev crosses each extended edge at most once. The number of these extended edges that ev crosses is $O(r)$.

2.2 Stage 2

In this stage tr has seen ev but ev has not yet begun to escape. The analysis of this stage (omitted here) utilizes optimal strategies for different cases.

2.3 Stage 1

We enumerate all possible $VP(ev)$ where tr can see ev. Then we apply the algorithm for stages 2 and 3. We choose the solution that maximizes the total time ev is seen.

While the results discussed here are for the (rather unnatural) model of rectangle vision within the polygon, the talk includes a discussion of the case of ordinary visibility, where the structure of an optimal solution is significantly more complex.

References

[1] S. Bhattacharya, S. Candido, and S. Hutchinson. Motion strategies for surveillance. In *Robotics: Science and Systems*, 2007.

[2] A. Efrat, H. H. González-Banos, S. G. Kobourov, and L. Palaniappan. Optimal strategies to track and capture a predictable target. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3789–3796. IEEE, 2003.

[3] C. Worman and J. M. Keil. Polygon decomposition and the orthogonal art gallery problem. *International Journal of Computational Geometry & Applications*, 17(02):105–138, Apr. 2007.