Guarding in a Simple Polygon[*]
(Extended Abstract)

Bor-Kuan Lu[†] Fang-Rong Hsu [‡] Chuan Yi Tang [§]

May 11, 1998

Guarding in a simple polygon was motivated by art gallery problems. A *diagonal guard* is a guard capable of moving along an edge or an internal diagonal in a polygon. A *chord guard* is a guard capable of moving along a chord wholly contained in a simple polygon. A *line guard* is a guard capable of moving along a line segment wholly contained in a polygon.

As a special case of the art gallery problem, we can consider the problem of determining whether a single guard can guard a polygon. A polygon which can be guarded by a diagonal guard, a chord guard, or a line guard is called *diagonal-visible*, *chord-visible*, or *line-visible*, respectively. Kim, Shin and Chwa [1] presented an $O(n \log n)$ algorithm for finding all diagonals of a simple polygon P on which a diagonal guard lies. Das, Heffernan and Narasimhan [2] presented a linear time algorithm for finding all weakly-visible chords of a polygon.

In the optimization problems, Chen [3] presented a linear time algorithm for finding the shortest weakly visible subedge. Das and Heffernan [4] presented a linear time algorithm for finding the shortest illuminating line segment in a polygon. But there was no paper talked about finding the shortest guard diagonal. Kim, Shin and Chwa [1] left the problem of finding the shortest guard diagonal as an open problem. In this paper, we talk about the algorithm of finding the shortest guard diagonal, the longest guard diagonal and the longest guard chord. Note that the longest guard chord is also the longest guard line (i.e. the longest illuminating line segment) in a polygon. In this paper, we shall propose algorithms for the following three problems:

1. Find the shortest guard diagonal in $O(n\alpha(n))$ time.

2. Find the longest guard diagonal in linear time.

3. Find the longest guard chord in linear time.

Note that $\alpha(n)$ is the inverse Ackermann's function. We use the matrix-searching technique [5], and some fairly basic concepts from computational geometry.

Let $\mathcal{P}$ be a *simple polygon* specified by a list $(v_0, v_1, \ldots, v_{n-1})$ of n vertices in the counter-clockwise order on the boundary. The *edge* of $\mathcal{P}$ joining v_i and v_{i+1} is denoted by $e(v_i, v_{i+1})$, with convention that $v_n = v_0$. $Bd(\mathcal{P})$ and $Int(\mathcal{P})$ denote the entire boundary and the interior of $\mathcal{P}$, respectively so that $\mathcal{P} = Int(\mathcal{P}) \cup Bd(\mathcal{P})$.

Before giving an overview of the algorithm for finding the longest guard diagonal and shortest guard diagonal, we describe the peculiar output of the Kim et al. [1] for computing all weakly-visible diagonals of a polygon. The algorithm in [1] generate n pairs of the form $(\alpha_i, \beta_i, guard - ch(v_i))$. Note that

[*]Support in part by NSC grant NSC-97-2213-E-127007

[†]Dept. of Computer Science, National Tsing Hua University, Taiwan, ROC. bklu@cs.nthu.edu.tw

[‡]Dept. of Accounting, Providence University, Taiwan, ROC. frhsu@acc.pu.edu.tw

[§]Dept. of Computer Science, National Tsing Hua University, Taiwan, ROC. cytang@cs.nthu.edu.tw

$guard - ch(v_i) = ch(b_i, e_i)$ and every diagonal $\overline{v_i, e}$, $e \in \{\alpha_i, \beta_i, guard - ch(v_i)\}$, is a weakly-visible diagonal in a polygon.

We define a $n \times n$ matrix $\mathcal{D}$ such that

$$\mathcal{D}_{ij} = \begin{cases} d(v_i, v_j) & \text{if } v_i \in guard - ch(v_i) \\ & \text{and } i < j \ , \\ blank & \text{otherwise} \end{cases}$$

Algorithm : Find the shortest (the longest) guard diagonal
Input : A simple polygon $\mathcal{P}$
Output: The shortest (the longest) guard diagonal
Begin

1. Generate n pairs of $(\alpha_i, \beta_i, guard - ch(v_i))$ if exists.
2. If there does not exist any weakly-visible diagonal, then there is no shortest guard diagonal. Otherwise, do the following steps.
3. Find the minimum (resp. maximum) amount all α_is and β_is.
4. Find the minimum (resp. maximum) elements in matrix $\mathcal{D}$.
5. Pick the minimum (resp. maximum) one from Step 3 and Step 4.

End.

As the above, we describe the peculiar output of the Das et al. [2] for computing *all weakly-visible chord* of a polygon. The algorithm in [2] generate $k = O(n)$ pairs of the form (A_i, B_i, x, y) Note that for the rest of this paper k will be used to denote the number of pairs output by the algorithm [2]. We now describe A_i and B_i. Here A_i is a subedge on $Bd(\mathcal{P})$ and it partition $Bd(\mathcal{P})$ contiguously in counter-clock direction. we call A_i *basic interval*. B_i is a polygonal chain on $\mathcal{P}$ such that B_i is the union of adjacent basic intervals. Every line segment joining a point on A_i and a point on B_i forms a weakly-visible chord. It may also be noted that each of the A_is are disjoint line segments, while the B_is are possibly overlapping polygonal chains.

We define a $k \times k$ matrix $\mathcal{LC}$ such that

$$\mathcal{LC}_{ij} = \begin{cases} max\{d(a,b) | a \in A_i \text{ and } b \in A_j\} \\ \text{if } A_j \in B_i \text{ and } i < j \ , \\ blank \\ \text{otherwise} \end{cases}$$

The non-blank elements in matrix $\mathcal{LC}$ are the length of the longest guard chord between A_i and A_j.

Algorithm : Find the longest guard chord
Input : A simple polygon $\mathcal{P}$
Output: The longest guard chord.
Begin

1. Generate k pairs of (A_i, B_i, x, y) if exists.
2. Find the maximum elements in matrix $\mathcal{LC}$.

End.

References

[1] S.Y. Shin S.H. Kim and K.Y. Chwa. Efficient algorithms for solving diagonal visibility problems in a simple polygon. *International Journal of Computational Geometry & Applications*, 5:433–458, 1995.

[2] G. Das, P.J. Heffernan, and G. Narasimhan. Finding all weakly-visible chords of a polygon in linear time. *Nordic Journal of Computing*, 1(4):433–457, 1994.

[3] D.Z. Chen. Optimally computing the shortest weakly visible subedge of a simple polygon. *Journal of Algorithms*, 20:459–478, 1996.

[4] G. Das and P.J. Heffernan. Optimal linear-time algorithm for the shortest illuminating line segment in a polygon. In *Proc. 10th Annual ACM Symp. on Computational Geometry*, pages 259–268, 1994.

[5] M. Klawe. Superlinear bounds for matrix searching problems. *Journal of Algorithms*, 13:55–78, 1992.