Computing the Dent Diameter of a Polygon

(Extended Abstract)

Peter Yamamoto*
Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1
pjyamamoto@barrow.uwaterloo.ca

1 Introduction

The link distance between two points is a measure of the minimum number of turns necessary to navigate from one point to another in the presence of obstacles. Minimum link paths in polygons, where the boundary of the polygon represents the obstacles, were extensively investigated by Suri [Sur87]. Suri motivated the problems by the desire to minimize the cost of turning in areas such as in motion planning where the cost of turning a robot is relatively high compared to traveling in a straight line; in broadcasting type problems, where it requires some type of relay to turn a signal; and in the Space Factory Problem where, again, turning requires more energy than moving in a straight line.

Suri also considered the link diameter problem: finding the maximum number of segments in any minimum link path of a given polygon. The diameter is of interest because it provides a measure of the geometric complexity of the polygon. Suri [Sur87] gave an $O(n \log n)$ time $O(n)$ space algorithm for computing the link diameter of a simple polygon. de Berg [dB91] considered the orthogonal version of the problem in which all line segments are axis-parallel and gave a straightforward divide and conquer algorithm with the same time complexity. Nilsson and Schuierer [NS91b] improved this and gave the first optimal algorithm for the diameter problem on a non-trivial class of polygons.

As mentioned above, the diameter is of interest since it provides a measure of geometric complexity of the polygon. Nilsson and Schuierer [NS91b] suggest that the link diameter gives a classification on the amount of *winding* in the polygon. Although the link diameter does provide a measure of turns, it can be artificially inflated without changing the general shape of the polygon. The link diameter does not provide information about the notion of winding of a polygon in the sense given by Chazelle and Incerpi [CI83] (sinuosity) or Sack [DS85] (winding). The **dent diameter** of a polygon provides more information than the link diameter about the shape complexity

*Supported in part by the Information Technology Research Centre of Ontario.

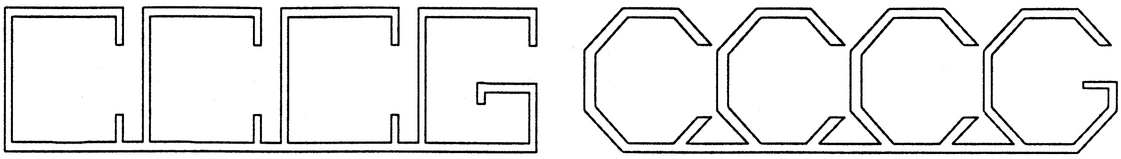

Figure 1: The polygon on the left has a link diameter of 7, a staircase diameter of 4. and a dent diameter of 5. The polygon on the right has a has a link diameter of 11, a staircase diameter of 6, and a dent diameter of 6.

of the polygon since it is a measure of the type of turning required.

Wood and Yamamoto [WY93], introduced the notion of dent distance, or visibility, and gave algorithms for efficiently solving some dent visibility problems. One of the motivations for studying dent distance is that it can provide more geometric information than the link distance function. In this paper, we continue the investigation of dent and staircase visibility and provide an algorithm for computing the dent diameter of a polygon based upon the algorithm given by de Berg [dB91] (the staircase diameter may also be solved in an analogous manner). Unfortunately, the geometric properties used by Nilsson and Schuierer [NS91b] to optimize the link algorithm cannot be applied directly to obtain an optimal dent diameter algorithm.

First, we review the notion of **dent length** and the divide and conquer approach for link diameter used by de Berg [dB91] and Nilsson and Schuierer [NS91b]. We then outline the results required to show that the general diameter algorithm may be applied to the dent diameter problem to compute the diameter in $O(n \log n)$ time.

2 Dent distance

In a simple orthogonal polygon, a **dent edge** is any edge with two reflex vertices. The notion of a dent in restricted orientation polygons was used by Culberson and Reckhow [CR89] and Bremmer and Shermer [BS92] in the solution to some minimal covering problems. The former used the notion of dents to characterize polygons for which a solution could be computed.

We adopt the convention that the polygon is labeled in a clockwise orientation; hence, an alternate definition for a dent edge is an edge with two left turns. Wood and Yamamoto [WY93] extended the notion of a dent in a polygon to a dent in a path. In this case, a **dent segment** is a segment of the path such that its vertices are both right turning or both left turning.

The **dent length** of a path is the number of dent segments in the path. We define the **orthogonal dent distance**, or simply dent distance, of two points as the minimum dent length over all orthogonal paths connecting the two points. An **orthogonal staircase path**, or simply staircase, is an orthogonal path such that the path is monotone with respect to the axes [SRW91]. We define the **staircase length** of a path as the minimum number of staircases joined at their endpoints which decompose the path. We define the **staircase distance** of a path between two points as the minimum staircase length over all orthogonal paths between the two points.

Except for 0-dent and 1-staircase length paths, the dent-length and staircase-length of a path are, in general, not equal. Note that there may be one or two dents in between two consecutive staircases. There are two dents if the staircases are oppositely oriented; otherwise there is just one dent. In Figure 1 the general shape of the two polygons is the same. However, we can refine the edges of the polygon with shorter and shorter segments, artificially inflating the link diameter as high as we want. The difference between the staircase and link diameters in the polygons above is due to the fact that we add a new turn which consists of two oppositely oriented staircases; however, further edge refinement would not increase the dent diameter value.

We consider the dent distance version of the following fundamental visibility problem: Given an orthogonal polygon, compute the (orthogonal) link diameter.

3 Link Diameter

The link diameter algorithm used by de Berg [dB91] and Nilsson and Schuierer [NS91b] is based on a straight forward divide and conquer approach. First the polygon P is split into two approximately equal sized subpolygons P_1 and P_2 by a chord c. Now, observe that the diameter of P is determined by one of three cases. The diameter is determined by the distance between two points in P_1, or between two points in P_2, or between one point in P_1 and one point in P_2. The algorithm outline is then as follows.

1. If P is a rectangle, then $D(P) = 2$.
2. Otherwise find a chord c dividing P into subpolygons P_1 and P_2 and such that $|P_1|, |P_2| \leq 3n/4 + 2$.
3. Compute $D(P_1)$ and $D(P_2)$ recursively.
4. Compute $M = \max\{d(v_1, v_2) | v_1 \in P_1, v_2 \in P_2\}$.
5. Let $D(P) = \max\{M, D(P_1), D(P_2)\}$.

Note that the algorithm outline is independent of the distance measure.

The existence of such a chord c in an orthogonal polygon is proven by the Orthogonal Polygon Cutting Theorem [dB91]. The dividing chords c_i allow us to represent the partition of P in a binary tree structure T. The root node represents P. If P is a rectangle then T consists of a single node. Otherwise, for each internal node t representing a subpolygon P_i, there exists a dividing chord c_i which splits the polygon P_i into two sub-polygons represented by its two children. The balancing of the dividing chords c_i means that the height of the tree is $O(\log n)$. Hence the running time of the algorithm is determined by the complexity of determining c and the cost of computing M.

de Berg [dB91] showed that c can be found in linear time with respect to the polygon. In order to compute M we do not want to consider too many pairs of points. de Berg [dB91] showed that it suffices to consider distances between vertices only. The following lemma establishes the equivalent property for dent distances.

Lemma 3.1 *Given an orthogonal polygon P and any two points p_1, $p_2 \in P$, there exists at least one vertex v such that the dent distance $d(p_1, p_2) \leq d(p_1, v)$.*

This lends itself to a brute force algorithm by inspecting vertex pairs. However, de Berg showed that M can be computed in linear time with respect to the size of P_1 and P_2.

4 Dent Parameterization

In this section we briefly outline the main points of the algorithm which are dependent on the distance function and give their dent equivalents. The focus of the problem is now on Step 4, computing M, the maximum distance $d(v, w)$ over all $v \in P_1$ and $w \in P_2$.

The main idea of the algorithm is to record, for each vertex v, a segment on c, which represents the points reachable by v in distance $d(c, v) = \min_{x \in c} d(x, v)$. The segments of two vertices v and w on different sides of c are then used to determine the extra cost (to the path from v to w) of joining the points with a path that crosses c. This is done by noting the relationship between the segments on c. By considering segments for each vertex on c at the same time we hope to be able to determine a furthest pair without doing pairwise comparisons.

First, we note that the segment for one vertex is connected. Second, note that one endpoint of the segment is an endpoint of c since the path from v to c can continue along c in a staircase direction without increasing the dent length of the path. For a particular point v we represent its segment on c by its endpoints v^1 and v^2, where v^2 is the endpoint coinciding with an endpoint of c. v^1 is chosen to maximize the (Euclidean) length of the segment and is actually the same as one of the endpoints computed in [dB91]. We say that c^2 determines the staircase orientation of the path from v to c.

Lemma 4.1 *Let c divide P into two subpolygons P_1 and P_2 such that $v \in P_1$ and $w \in P_2$, and let $d(v,c) = d_v$ and $d(w,c) = d_w$. Then we have*

$$d(v,w) = d_v + d_w + \Delta$$

where

$$\Delta = \begin{cases} 0 & \text{if } v^2 \neq w^2 \wedge v^1 \in (w^1, w^2) \\ 1 & \text{if } v^2 = w^2 \\ 2 & \text{otherwise.} \end{cases}$$

Since this is the basis of the next result we explain the three cases. In the first case, the paths from v and w arrive at c with opposite staircase orientations ($v^2 \neq w^2$) from different sides. Since $v^1 \in (w^1, w^2)$, this means that the two staircases have a point in common and hence can be joined together without forming a dent. In the second case, the paths arrive with the same orientation. This means that the staircases go up and then down (or down then up) as they cross c, and hence incur 1 dent. The final case has the staircases in opposite orientation but this time they need a new edge to join them up. This edge forms causes the last link of each staircase to become a dent edge and hence increases the total dent length by 2.

Next, we formulate the computation of M in a similar manner, extending the case analysis to 3 possible cases for all the segments of all the vertices. Let $d_1 = \max\{d(v,e)\ v \in P_1\}$ and $d_2 = \max\{d(w,e)\ w \in P_2\}$. Next, let $P_i^k = \{u \text{ is a vertex of } P_i | d(u,c) = k\}$ for $i = 1,2$ denote the set of vertices of P_i at distance k from c. Note that $d_1 + d_2 \geq M \leq d_1 + d_2 + 2$ and that to achieve the upper bound of $d_1 + d_2$ the two vertices must come from $P_1^{d_1}$ and $P_2^{d_2}$ respectively. However, if $M < d_1 + d_2 + 2$ then a vertex v_1 which determines the maximum distance might be at distance $d_1 - 1$. From Lemma 4.1 we then have the following.

Lemma 4.2

$$M = d_1 + d_2 + \Delta$$

where

$$\Delta = \begin{cases} 2 & \text{if } \exists\, v \in P_1^{d_1} \text{ and } w \in P_2^{d_2} \\ & \text{such that } (v^1, v^2) \cap (w^1, w^2) = \emptyset \\ 1 & \text{else if } \exists\, v \in P_1^{d_1} \text{ and } w \in P_1^{d_2} \\ & \text{such that } v^2 = w^2 \\ & \text{or} \\ & \exists\, v \in P_1^{d_1 - 1} \text{ and } w \in P_1^{d_2} \\ & \text{such that } (v^1, v^2) \cap (w^1, w^2) = \emptyset \\ & \text{or} \\ & \exists\, v \in P_1^{d_1} \text{ and } w \in P_1^{d_2 - 1} \\ & \text{such that } (v^1, v^2) \cap (w^1, w^2) = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Without loss of generality suppose c is vertical and c_1 is higher than c_2. We first consider vertices $v \in P_1^{d_1}$ such that $v^2 = c^1$ and vertices $w \in P_1^{d^2}$ such that $w^2 = c^2$. The first condition is checking that no intervals from vertices of $P_1^{d_1}$ intersect intervals from vertices of $P_2^{d_2}$ and may be rewritten as:

$$\min\{v^1 | v \in P_1^{d_1} \text{ and } v^2 = c^1\} >$$
$$\max\{w^1 | w \in P_2^{d_2} \text{ and } w^2 = c^2\}.$$

To check the other candidates for this case just switch c^1 and c^2 in the equations.

The next condition involves three sets of vertices. First we check vertices $v \in P_1^{d_1}$ and $w \in P_2^{d_2}$ to see if a pair of segments intersect at an endpoint of c:

$$\begin{aligned} \max_{v\in P_1^{d_1}} v^2 &= \max_{w\in P_1^{d_1}} w^2 = c^1 \\ or & \\ \max_{v\in P_1^{d_1}} v^2 &= \max_{w\in P_1^{d_1}} w^2 = c^2. \end{aligned}$$

If neither of these conditions hold then we need to check vertices from $v \in P_1^{d_1-1}$ and $w \in P_2^{d_2}$. In this case we are testing for the same condition as in the first case, to see if the path between any two vertices gains two dents by crossing c. Similarly we check for $v \in P_1^{d_1}$ and and $w \in P_2^{d_2-1}$. Otherwise the maximum distance between vertices in P_1 and P_2 is $d_1 + d_2$. All these conditions can be checked in linear time. Hence Step 4 of the algorithm takes at most linear time, providing the following result.

Theorem 4.3 *The dent diameter of an orthogonal polygon with n vertices can be computed in $O(n \log n)$ time.*

5 Conclusion

We have noted the interesting developments of the orthogonal link diameter problem and extended those results to another distance function, to solve the dent diameter problem in $O(n \log n)$ time. The dent diameter is of interest for its additional measure of shape complexity of a polygon. Since the orthogonal link diameter problem may be solved in optimal linear time we suspect that a different approach should also allow the link diameter to be computed in linear time. An interesting problem is to generalize these results for restricted orientation polygons.

Also related to this problem is the center problem: find the set of points such that the distance to their furthest neighbor is minimized. Lenhart *et al.* [LPS+87] gave an algorithm for computing the link center of a simple polygon in $O(n^2)$ time and $O(n^2)$ space. Subsequently Djidjev *et al.* [DLS89] and Ke [Ke89] both provided $O(n \log n)$ time algorithms for computing the link center. Nilsson and Schuierer [NS91a] again considered the orthogonal version of this problem and gave an optimal $O(n)$ time algorithm. Once again the algorithms may be adapted to solve the staircase and dent distance versions of the center problem however a direct application does not produce an optimal time algorithm.

Acknowledgements

The author would like to thank Sven Schuierer and Derick Wood for valuable discussions and continuing research on this topic.

References

[BS92] David Bremmer and Thomas Shermer. Properties and applications of restricted-orientation point visibility graphs. Manuscript, School of Computing Science, Simon Fraser University, 1992.

[CI83] B. Chazelle and J. Incerpi. Triangulating a polygon by divide and conquer. In *Proc. 21st Allerton Conf. Commun. Control Comput.*, pages 447–456, 1983.

[CR89] J. Culberson and R. A. Reckhow. Orthogonally convex coverings of orthogonal polygons without holes. *J. Comput. Syst. Sci.*, 39:166–204, 1989.

[dB91] M. de Berg. On rectilinear link distance. *Comput. Geom. Theory Appl.*, 1(1):13–34, July 1991.

[DLS89] H. N. Djidjev, A. Lingas, and J. Sack. An $O(n \log n)$ algorithm for computing the link center of a simple polygon. In B. Monien and R. Cori, editors, *Proc. 6th Sympos. Theoret. Aspects Comput. Sci.*, volume 349 of *Lecture Notes in Computer Science*, pages 96–107. Springer-Verlag, 1989.

[DS85] J. A. Dean and J.-R. Sack. Efficient hidden-line elimination by capturing winding information. In *Proc. 23rd Allerton Conf. Commun. Control Comput.*, pages 496–505, 1985.

[Ke89] Y. Ke. An efficient algorithm for link-distance problems. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 69–78, 1989.

[LPS+87] W. Lenhart, R. Pollack, J.-R. Sack, R. Seidel, M. Sharir, S. Suri, G. T. Toussaint, S. Whitesides, and C. K. Yap. Computing the link center of a simple polygon. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 1987.

[NS91a] B. J. Nilsson and S. Schuierer. An optimal algorithm for the rectilinear link center of a rectilinear polygon. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pages 249–260. Springer-Verlag, 1991.

[NS91b] Bengt J. Nilsson and Sven Schuierer. Computing the rectilinear link diameter of a polygon. In *Computational Geometry — Methods, Algorithms and Applications: Proc. Internat. Workshop Comput. Geom. CG '91*, volume 553 of *Lecture Notes in Computer Science*, pages 203–215. Springer-Verlag, 1991.

[SRW91] Sven Schuierer, Gregory J. E. Rawlins, and Derick Wood. A generalization of staircase visibility. In *Computational Geometry — Methods, Algorithms and Applications: Proc. Internat. Workshop Comput. Geom. CG '91*, volume 553 of *Lecture Notes in Computer Science*, pages 277–287. Springer-Verlag, 1991.

[Sur87] S. Suri. *Minimum link paths in polygons and related problems.* Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1987.

[WY93] D. Wood and P. Yamamoto. Dent and staircase visibility. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 297–302, Waterloo, Canada, 1993.