Computing the Rectilinear Link Diameter of a Polygon*

Bengt J. Nilsson†  Sven Schuierer†

Abstract

The problem of finding the diameter of a simple polygon has been studied extensively in recent years. $O(n \log n)$ time upper bounds have been given for computing the geodesic diameter and the link diameter for a polygon.

We consider the rectilinear case of this problem and give a linear time algorithm to compute the rectilinear link diameter of a simple rectilinear polygon. To our knowledge this is the first optimal algorithm for the diameter problem of non-trivial classes of polygons.

1 Introduction

In many versions of motion planning problems the large cost involves making turns. An example is broadcasting a radio signal through a beam. At particular points relay stations must be erected to reflect the beam in a new direction. This leads to a cost measure for paths which is commonly known as the link metric.

The problem of finding link optimal paths between points inside a simple polygon has been solved by Suri [Sur87]. He gives a linear time algorithm for this problem. In addition he shows that the link diameter, the furthest link distance between any two points in a polygon, can be computed in $O(n \log n)$ time and linear storage ($n$ will in the following always denote the number of polygon edges). A similar result is obtained by Ke [Ke89].

In many cases we may want to restrict the paths to be rectilinear, i.e., all segments of the path are axis parallel. This particular restriction has been studied for simple rectilinear polygons by de Berg [B89]. He devises a data structure which allows to compute, given two query points inside the polygon, the shortest path between the points in $O(\log n + l)$ time, where $l$ is the number of links of the path. The data structure can be constructed in $O(n \log n)$ time and requires $O(n \log n)$ storage. Furthermore, he shows how to compute the rectilinear link diameter in $O(n \log n)$ time with a simple divide and conquer approach.

In the more general case when the obstacles do not form the boundary of a simple polygon but instead are just rectilinear line segments in the plane, a generalization of the shortest path problem has been studied by de Berg et al. [BKNO90].

In this paper we give a linear time algorithm for computing the rectilinear link diameter of a simple rectilinear polygon. To our knowledge this is the first optimal algorithm found for the diameter problem of non-trivial classes of polygons. The algorithm is based on a divide and conquer algorithm presented in [B89]. The idea is to split the polygon into two subpolygons of approximately equal size, compute the diameter of the two subpolygons recursively, and to compute the longest distance between any pair of vertices in different subpolygons. The maximum of these three values is the diameter. This algorithm runs in $O(n \log n)$ time. In order to reduce the running time to linear it is possible to show that one only needs to recur on one of the subpolygons. For the other subpolygon one can show that the diameter is either smaller than the diameter of the full polygon or the diameter of the subpolygon can be computed explicitly without recursion and in linear time.

The paper is organized as follows. In the next section we state our definitions and give some preliminary results for simple rectilinear polygons. We show in Section 3 how to compute the link

---

*This work was supported by the Deutsche Forschungs Gemeinschaft under Grant No. O1 64/5-4.
†Institut für Informatik, Universität Freiburg, Rheinstr. 10-12, D-7800 Freiburg, Fed. Rep. of Germany.
diameter in optimal time. In Section 4 we present a simple algorithm to compute an approximate
link center, i.e., a point from which all other points are reachable using at most \( R(P) + 1 \) links, in
linear time (\( R(P) \) is the rectilinear link radius of polygon \( P \)).

2 Definitions

Let \( P \) be a Jordan curve consisting of \( n \) axis parallel line segments such that no two consecutive
segments are collinear. We define a **simple rectilinear polygon** \( P \) to be the union of \( P \) and its interior.

A (rectilinear) path \( \mathcal{P} \) is a curve that consists of (a finite number of) axis parallel line segments inside
\( P \). The length of \( \mathcal{P} \), denoted by \( \lambda(\mathcal{P}) \), is the number of line segments it consists of. From now on
we will only consider rectilinear polygons and rectilinear paths in a polygon \( P \). Hence, whenever
we talk of polygons, we mean simple rectilinear polygons and whenever we talk of paths, we mean
rectilinear paths in \( P \).

Let \( p \) and \( q \) be two points in a polygon \( P \) and \( e \) be an axis parallel line segment in \( P \). The
(rectilinear) link distance between \( p \) and \( q \), denoted by \( d(p, q) \), is defined as the length of the shortest
path connecting \( p \) and \( q \). We say a polygonal path \( \mathcal{P} \) from \( p \) to line segment \( e \) is **admissible** if it is
rectilinear and the last link of \( \mathcal{P} \) is orthogonal to \( e \). We define the link distance of \( p \) and \( e \), again
denoted by \( d(p, e) \), to be the length of the shortest admissible path from \( p \) to a point of \( e \).

The (rectilinear) link diameter denoted \( D(P) \) is defined as the maximum rectilinear link distance
between any two points in \( P \). The diameter gives a classification on the amount of winding in
the polygon. There is an interesting relationship between the link diameter and the (rectilinear)
link radius of \( P \), denoted by \( R(P) \), which is defined as the minimum integer \( k \) for which there
is some point in \( P \) from which all other points can be reached by \( k \) links. It can be shown that
\[ \lceil D(P)/2 \rceil \leq R(P) \leq \lceil D(P)/2 \rceil + 1 \] and that these bounds are tight. We will make use of this fact
in Section 4 to compute a point that can reach all points in \( P \) with at most \( R(P) + 1 \) links. Such a
point can viewed as an approximation to the link center of \( P \). The **link center** of \( P \) is the set of all
points that can reach all other points using at most \( R(P) \) links.

In the rest of the paper the letter \( v \) with various subscripts and superscripts will always denote
vertices of \( P \). In the same way the letters \( p \), \( q \), and \( r \) denote points of \( P \).

In the following lemma it is proven that there is always a vertex which is a furthest neighbour
of a point of the polygon. This fact is used to discretize the problem, i.e., our link diameter algorithm
only needs to concern itself with the vertices of \( P \).

**Lemma 2.1** For any two points \( p, q \in P \), there is a vertex \( v \) such that \( d(p, q) \leq d(p, v) \).

**Proof:** Let \( k = d(p, q) \) and let \( \mathcal{P} \) be a shortest path from \( p \) to \( q \). Let \( r \) be the point of intersection
between the boundary of \( P \) and the extension of the last link \( l \) of \( \mathcal{P} \). Let \( \mathcal{P}' \) be a shortest path from
\( p \) to \( r \) and suppose furthermore that \( \lambda(\mathcal{P}') < k \). Obviously, \( q \) can not be a point on \( \mathcal{P}' \). If the last
link of \( \mathcal{P}' \) is collinear with \( l \), \( q \) can be reached from \( p \) with less than \( k \) links, a contradiction. On the
other hand, if \( l \) is orthogonal to the last link of \( \mathcal{P}' \), then follow the two paths \( \mathcal{P} \) and \( \mathcal{P}' \) from \( r \) until
they join again. The path traced like this bounds a simple polygon \( P' \) interior to \( P \). The point \( q \) is
obviously a point on the boundary of \( P' \). Also, since \( \mathcal{P} \) is a shortest path, the chord \( c \) of \( P' \) through

![Figure 1: One of \( v \) and \( v' \) is at least as far from \( p \) as \( q \) is.](image-url)