An Optimal Algorithm for the Rectilinear Link Center of a Rectilinear Polygon

Bengt J. Nilsson† Sven Schuierer‡

Abstract

The link distance between two points in a polygon $P$ is defined as the minimum number of line segments inside $P$ needed to connect the two points. The link center of a polygon $P$ is the set of points in $P$ which minimizes the maximum link distance to all points in $P$. The problem of finding the link center of a simple polygon with $n$ edges has been studied extensively in recent years. Several $O(n \log n)$ time algorithms have been given for this problem.

We consider the rectilinear case of this problem and give a linear time algorithm to compute the rectilinear link center of a simple rectilinear polygon.

1 Introduction

The computation of shortest paths is an important problem in computational geometry. In some applications the cost of making a turn overshadows the cost of traversing a straight path by far. 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, a costly operation, whereas the transmission of the radio signal is cheap in relation. This leads to a cost measure for paths which is commonly known as the link metric.

The problem of finding a shortest path between two points inside a simple polygon of $n$ edges in the link metric has been studied by Suri [10]. He gives

---

*This work was supported by the Deutsche Forschungs Gemeinschaft under Grant No. Ot 64/5-4.
†Department of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden
a linear time algorithm for this problem. The link center of a polygon $P$ is
the set of points in $P$ which minimizes the maximum link distance to
all points in $P$. Lenhart et al. [6] obtained a quadratic time algorithm to
compute the link center of a simple polygon with $n$ edges that was later
improved to $O(n \log n)$ time by Djidjev et al. [3] and Ke [5].

Sometimes it is useful to restrict the paths to be rectilinear, i.e., all
segments of the path are axis parallel. In this case Das and Narasimhan [1]
present an $O(n \log n)$ time algorithm to compute a shortest rectilinear link
path between two points in a rectilinear polygon with rectilinear holes. In
simple rectilinear polygons de Berg [2] shows that there is a rectilinear path
which is shortest both in the Euclidean and the link metric. He constructs
a data structure that allows to compute, given two query points inside the
polygon, an $L_1$ and link shortest path between the points in $O(\log n +
l)$ time, where $l$ is the number of links of the path. The rectilinear link
diameter of a simple rectilinear polygon $P$ is the maximum rectilinear link
distance between any two points in $P$. In [2], de Berg also gives an $O(n \log n)$
algorithm to compute the rectilinear link diameter which was later improved

In the parallel setting the problems of computing a shortest rectilinear path
between two points and the rectilinear link center in a simple rectilinear
polygon have also been solved [4, 7, 8].

In this paper we give a linear time algorithm for computing the rectilinear
link center of a simple rectilinear polygon. The paper is organized as follows.
In the next section we state our definitions and give some preliminary results
for simple rectilinear polygons. Before presenting the actual algorithm for
the link center we need to develop two tools. The first one is a subroutine
which deals with computing the part of the link center inside a maximal
histogram of a polygon. It is presented in Section 3. The second tool is a
procedure to compute a chord that has vertices on both sides that are “far
away” from it. We call such a chord a central chord. Section 4 deals with
the construction of a central chord. Finally, in Section 5 we put everything
together and present the algorithm to compute the link center.

## 2 Definitions and Preliminary Results

Let $\mathcal{C}$ be a simple, closed 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 $\mathcal{C}$ and its interior. A (rectilinear)
path \( \mathcal{P} \) is a curve that consists of axis parallel line segments inside \( \mathcal{P} \). The \textit{length of} \( \mathcal{P} \), denoted by \( \lambda(\mathcal{P}) \), is the number of line segments it consists of. In this paper, when we talk of polygons, we mean simple rectilinear polygons and when we talk of paths, we mean rectilinear paths in a polygon.

Let \( p \) and \( q \) be two points in a polygon \( \mathcal{P} \) and let \( e \) be an axis parallel line segment in \( \mathcal{P} \). The \textit{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 the line segment \( e \) is \textit{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 \textit{(rectilinear) link eccentricity} of a point \( p \) is the maximum link distance between \( p \) and any point in \( \mathcal{P} \). We denote it by \( \varepsilon(p, \mathcal{P}) \). A point \( q \) with \( d(p, q) = \varepsilon(p, \mathcal{P}) \) is called a \textit{furthest neighbour} of \( p \). Similarly, if \( e \) is a line segment in \( \mathcal{P} \), we define the \textit{eccentricity} of \( e \) in \( \mathcal{P} \) as the maximum link distance between \( e \) and any point in \( \mathcal{P} \) and denote it by \( \varepsilon(e, \mathcal{P}) \).

The \textit{(rectilinear) link diameter}, denoted by \( D(\mathcal{P}) \), is defined as the maximum eccentricity of a point in \( \mathcal{P} \). The \textit{(rectilinear) link radius}, denoted by \( R(\mathcal{P}) \), is defined as the minimum eccentricity of a point in \( \mathcal{P} \). When no ambiguity can occur we will refer to the radius as \( R \). Finally, we define the \textit{(rectilinear) link center} \( \text{lc}(\mathcal{P}) \) as the set of points in \( \mathcal{P} \) that can reach all points using at most \( R \) links. Since it can be shown that there is always a furthest neighbour of a point that is a vertex \([9]\), the link center is also the set of points that can reach all \textit{vertices} using at most \( R \) links.

The next lemma establishes upper and lower bounds on the radius of a polygon.

**Lemma 2.1**

\[ \lfloor D(\mathcal{P})/2 \rfloor \leq R(\mathcal{P}) \leq \lceil D(\mathcal{P})/2 \rceil + 1 \]

**Proof:** The first inequality is easy to see by concatenating two paths to a point in the link center. The proof of second inequality is much more involved and therefore postponed to Section 4. \( \Box \)

As in some of the algorithms for the link center in a simple non-rectilinear polygon \([3, 6]\), our algorithm assumes a priori knowledge of the value of the radius. Since the link diameter can be found in linear time \([9]\), Lemma 2.1 implies that our algorithm can be applied in the following way. We first execute the algorithm with the value \( R(\mathcal{P}) = \lfloor D(\mathcal{P})/2 \rfloor \). If the algorithm produces a non-empty region, then we are done. Otherwise, \( R(\mathcal{P}) = \lceil D(\mathcal{P})/2 \rceil + 1 \) and we run the algorithm again with this value.
An interesting fact is that the rectilinear link center may be disconnected; see Figure 1a. In this example the diameter is 4 and the radius is 2.

A line segment \( c \) interior to \( \mathbf{P} \) is a chord if \( c \) is axis parallel, the end points of \( c \) intersect the boundary of \( \mathbf{P} \), and there is at least one point of \( c \) not on the boundary of \( \mathbf{P} \).

**Definition 2.1** Given a chord \( c \) it partitions the polygon into two subpolygons. A maximal histogram is the set of points in either subpolygon that can reach \( c \) with one rectilinear link; see Figure 1b. Note that a chord thus specifies two maximal histograms, one on each side of \( c \). In the following when we talk about histograms, we always mean maximal histograms. The chord \( c \) is called the base of the histogram. We also allow the base to be an edge or part of an edge of the polygon, in which case the base only specifies one maximal histogram.

A window is a maximal segment of the boundary of a histogram which is disjoint from the boundary of \( \mathbf{P} \). Note that the base of a histogram is also a window unless it coincides with an edge of \( \mathbf{P} \). When the base is horizontal and all interior points of histogram \( \mathbf{H} \) lie above the base, we will use the term west (east) edge to denote a vertical edge of \( \mathbf{H} \) that has the interior to the right (left) of the edge.

For an axis parallel line segment \( e \) and a point \( p \) in \( \mathbf{P} \), let \( e(p,d) \) denote the part of \( e \) that can be reached from \( p \) with an admissible path of
length at most \( d \). de Berg shows that, for any edge \( e \) of \( P \), the set of distances \( \{ d(v, e) \mid v \in P \} \) and the sets of subedges \( \{ e(v, d(v, e)) \mid v \in P \} \) and \( \{ e(v, d(v, e) + 1) \mid v \in P \} \) can be computed in linear time [2]. It is important to note that \( e(p, d) \) is a connected set, for any point \( p \) and any integer \( d \). Because of this property, we will often refer to the subedges as intervals.

For further reference we need the following lemmas from [2] and [9].

**Lemma 2.2 (de Berg)** [2] Let \( c \) be a chord which partitions \( P \) into two subpolygons such that \( p \) and \( q \) are points in different subpolygons and let \( d(p, c) = d_p \) and \( d(q, c) = d_q \). Then

\[
\begin{align*}
d(p, q) & = d_p + d_q - 1 & \text{when } c(p, d_p) \cap c(q, d_q) \neq \emptyset, \\
& \geq d_p + d_q & \text{otherwise}.
\end{align*}
\]

**Lemma 2.3 (Nilsson, Schuierer)** [9] Let \( e \) be an edge of polygon \( P \). If \( p \) and \( q \) are two points in \( P \) and \( d_p \) and \( d_q \) are two integers such that \( e(p, d_p) \) and \( e(q, d_q) \) are non-empty and \( e(p, d_p) \cap e(q, d_q) = \emptyset \), then \( d(p, q) \geq d_p + d_q - 1 \).

Sometimes we make use of the following slightly different formulation of Lemma 2.2.

**Corollary 2.1** If \( c \) is a chord which partitions \( P \) into two subpolygons such that \( p \) and \( q \) are points in different subpolygons, then \( d(p, c) \leq d(p, q) - d(q, c) + 1 \).

## 3 The Solution to a Special Case

In this section we establish a solution to a special case of the link center problem. Suppose we are given a polygon \( P \), its radius \( R \), and a maximal histogram \( H \) along an axis parallel chord of the polygon. We consider how to compute the part of the link center contained in \( H \).

Let \( c \) be a horizontal chord in \( P \) and \( H \) a maximal histogram of \( c \) with its interior above \( c \). We denote the region in \( H \) that can be reached with \( k \) links from all vertices in \( P \) by \( A_k(H) \). Clearly, the part of the link center contained in \( H \) equals \( A_R(H) \). Since we later need to compute the region of points in \( H \) that can be reached with \( k \) links from all vertices of \( P \) for values \( k \) different from \( R \), we show in the following how to compute \( A_k(H) \) in linear time (in the size of \( P \)) where \( k \) is an arbitrary integer.
If $w$ is a window of $\mathbf{H}$, we denote the subpolyon cut off by $w$ as $\mathbf{P}_w$ and the region in $\mathbf{H}$ that can be reached with at most $k$ links from all vertices in $\mathbf{P}_w$ by $\mathbf{A}_k(w)$. Clearly, the intersection of all regions $\mathbf{A}_k(w)$ equals $\mathbf{A}_k(H)$.

Therefore, we first consider how to compute $\mathbf{A}_k(w)$ for a single window $w$.

### 3.1 Computing $\mathbf{A}_k(w)$

In the following we assume that $w$ is a fixed west window of $\mathbf{H}$. We denote the distance of $w$ to its furthest neighbor in $\mathbf{P}_w$ by $d_w$.

First note that if $d_w \leq k - 3$, then all vertices in $\mathbf{P}_w$ can reach all points in $\mathbf{H}$ with $k$ links and if $d_w > k$, i.e., there is a vertex $v$ in $\mathbf{P}_w$ with $d(v, w) > k$, then no point in $\mathbf{H}$ can be reached from $v$ with $k$ links. Hence, we assume in the following that $k - 2 \leq d_w \leq k$.

If $v$ is a vertex in $\mathbf{P}_w$, we denote the region in $\mathbf{H}$ that can be reached with $k$ links from $v$ by $\mathbf{A}_k(v)$. Obviously, we have

$$\mathbf{A}_k(w) = \bigcap_{v \in \mathbf{P}_w} \mathbf{A}_k(v).$$

In fact, we only have to consider the vertices of $\mathbf{P}_w$ at distances $d_w$ or $d_w + 1$ to $w$.

**Lemma 3.1** If $v_1$ and $v_2$ are two vertices in $\mathbf{P}_w$ with $d(v_1, w) = d_w$ and $d(v_2, w) \leq d_w - 2$, then $\mathbf{A}_k(v_1) \subseteq \mathbf{A}_k(v_2)$.

**Proof:** Let $p$ be a point in $\mathbf{A}_k(v_1)$ and $\mathcal{P}$ a shortest path from $v_1$ to $p$ with $\lambda(\mathcal{P}) \leq k$. Since the link of $\mathcal{P}$ that intersects $w$ is at least the $d_w$-th link, we can choose a path from $v_2$ to $w$ of length $d_w - 2$ and connect it with two more links to $\mathcal{P}$ which yields a path of length at most $k$ from $v_2$ to $p$. □

If we define $\mathbf{A}_k(w, d)$ as the intersection of the regions $\mathbf{A}_k(v)$, where vertex $v$ is at distance exactly $d$ to $w$, then Lemma 3.1 can also be stated as follows.

**Corollary 3.1** The region $\mathbf{A}_k(w)$ is the intersection of $\mathbf{A}_k(w, d_w)$ and $\mathbf{A}_k(w, d_w - 1)$.

Since $\mathbf{A}_k(w, d_w)$ and $\mathbf{A}_k(w, d_w - 1)$ are monotone regions, as we will show later, their intersection $\mathbf{A}_k(w)$ can be computed in linear time. Hence, we only have to consider how to compute the two regions $\mathbf{A}_k(w, d_w)$ and $\mathbf{A}_k(w, d_w - 1)$ for $w$.

As we observed above the only non-trivial values for $d_w$ to be considered are in the range $[k - 2, k]$. Hence, by Corollary 3.1 we only have to show how to compute $\mathbf{A}_k(w, d)$, for $k - 3 \leq d \leq k$, where the value $k - 3$ is again uninteresting since the region $\mathbf{A}_k(w, k - 3)$ is the whole histogram $\mathbf{H}$. 6
3.1.1 $\delta$-Areas

In order to compute $A_k(v, d)$ with $k - 2 \leq d \leq k$, we need the notion of the $\delta$-area of an interval. If $I$ is an interval on the boundary of $\mathcal{P}$, then the $\delta$-area $a_\delta(I, \mathcal{P})$ of $I$ in $\mathcal{P}$ is the region of points in $\mathcal{P}$ that is reachable with an admissible path of $\delta$ links from $I$. Note that the 1-area of $I$ is just the maximal histogram of $I$ in $\mathcal{P}$. An example of the 1-, 2-, and 3-area of an interval is given in Figure 2a. Since we only consider $\delta$-areas in $\mathcal{H}$ in this section, we denote the $\delta$-area of an interval $I$ in $\mathcal{H}$ by $a_\delta(I)$.

**Lemma 3.2** If $v$ is a vertex of $\mathcal{P}_w$ with distance $d_v$ to $w$ and $I$ and $J$ are the intervals on $w$ that $v$ can reach with an admissible path of length $d_v$ resp. $d_v + 1$, i.e., $I = w(v, d_v)$ and $J = w(v, d_v + 1)$, then $A_k(v)$ is the union of the $(k - d_v)$-area of $I$ and the $(k - d_v)$-area of $J$ in $\mathcal{H}$.

**Proof:** Let $p$ be point in $A_k(v)$ and $\mathcal{P}$ a shortest path from $v$ to $p$. Furthermore, let $l$ be the first link of $\mathcal{P}$ that intersects $w$. If $l$ intersects $I = w(v, d_v)$, then the part of $\mathcal{P}$ from $w$ to $p$ consists of at most $(k - d_v + 1)$ links, the first one of which is orthogonal to $w$. Hence, $p$ is contained in the $(k - d_v + 1)$-area of $I$. A similar argument holds if $\mathcal{P}$ intersects $J \setminus I$. If $l$ intersects $w \setminus (I \cup J)$, then $l$ is at least the $(d + 2)$-nd link of $\mathcal{P}$. So we can choose a path $\mathcal{P}'$ from $v$ to $w$ consisting of $d_v$-links and connect it with two more links in $\mathcal{H}$ to $l$.
If \( p \) is contained in the \((k - d_v + 1)\)-area of \( I \), then there is an admissible path \( \mathcal{P}' \) from \( p \) to a point \( q \) on \( I \) consisting of \((k - d_v + 1)\) links. Since \( q \) is in \( I \), there is also an admissible path \( \mathcal{P} \) of length \( d_v \) from \( v \) to \( q \). Furthermore, since the last links of \( \mathcal{P} \) and \( \mathcal{P}' \) are both horizontal, we can join them at \( q \) to yield a path of length \( \lambda(\mathcal{P}) + \lambda(\mathcal{P}') - 1 = k \); see Figure 2b. A similar argument holds if \( p \) is contained in the \((k - d_v)\)-area of \( J \). □

Let \( \mathcal{V}_d(w) \) denote the set of vertices in \( \mathcal{P}_w \) at distance \( d \) to \( w \). In order to compute \( A_k(w, d) \) we need to compute the intersection of all sets which are the union of the \((k - d + 1)\)-area \( a_{k-d+1}(w(v,d)) \) of \( w(v,d) \) and the \((k - d)\)-area \( a_{k-d}(w(v,d+1)) \) of \( w(v,d+1) \), where \( v \) is a vertex in \( \mathcal{V}_d(w) \) and \( k - 2 \leq d \leq k \).

### 3.1.2 Computing \( A_k(w, k) \)

First we consider the case \( d = k \). If \( v \) is in \( \mathcal{V}_k(w) \), then \( A_k(v) \) is the union of the 1-area of \( w(v,k) \) and the 0-area of \( w(v, k + 1) \) by Lemma 3.2. Since the 0-area of an interval is empty, \( A_k(w, k) \) is just the intersection of all the 1-areas of the intervals \( w(v, k) \), with \( v \) in \( \mathcal{V}_k(w) \). It is easy to see that the intersection of the 1-areas is the 1-area of the intersection of the intervals \( w(v, k) \) with \( v \) in \( \mathcal{V}_k(w) \). This is straightforward computable in time \( O(H + |P_w|) \). Note that given an integer \( d \) the set of intervals \( w(v,d) \), for every vertex \( v \in P_w \), can be computed in time \( O(|P_w|) \) [2].

### 3.1.3 Computing \( A_k(w, k - 1) \)

Next we look at the case \( d = k - 1 \). In order to compute the intersection of all sets of the form \( a_2(w(v, k - 1)) \cup a_1(w(v, k)) \) where \( v \) is a vertex in \( \mathcal{V}_{k-1}(w) \) we need the following observation which is easy to prove.

**Lemma 3.3** Let \( I_1 \) and \( I_2 \) be two intervals on \( w \) with bottommost points \( p_1 \) and \( p_2 \). If \( p_1 \) is above \( p_2 \) and \( \delta \geq 2 \), then the \( \delta \)-area of \( I_1 \) in \( H \) is contained in the \( \delta \)-area of \( I_2 \) in \( H \).

Let \( \mathcal{I}_d \) be the set of intervals \( w(v,d) \) with \( v \in \mathcal{V}_d(w) \). We say an interval \( I \) in \( \mathcal{I}_d \) is a *topmost* interval in \( \mathcal{I}_d \) if the bottommost point of \( I \) is above or as high as the bottommost points of all other intervals in \( \mathcal{I}_d \).

Let \( I \) be a topmost interval in \( \mathcal{I}_{k-1} \) and \( H_w \) the maximal histogram of \( w \) in \( H \). By the above lemma the 2-area of \( I \) is contained in \( A_k(w, k - 1) \). Furthermore, all the points of \( A_k(w, k - 1) \) outside the 2-area \( I \) are contained in the 1-area of \( w \), that is, in \( H_w \).
Since the 2-area of $I$ can easily be computed in linear time, we show in the following only how to compute the part of $A_k(w, k-1)$ in $H_w$. Note that $H_w$ is a horizontally and vertically monotone histogram whose left boundary and lower boundary consist of just one single line segment. Clearly, $H_w$ can be computed in linear time.

We apply a plane sweep approach in $H_w$ from $w$ towards the right to compute $A_k(w, k-1) \cap H_w$. The event points are the vertical edges of $H_w$. Note that a vertical edge always belongs to the upper boundary of $H_w$. The algorithm outputs a subpyramid $Y$ of $H_w$ that equals the part of $A_k(w, k-1)$ in $H_w$. $Y$ is represented by its upper and lower boundaries $U$ and $L$ both of which form an $xy$-montone staircase.

The algorithm keeps a list $L_{k-1}$ of the intervals $w(v, k-1)$, for all vertices $v \in V_{k-1}(w)$. $L_{k-1}$ is sorted by the lower end points of the intervals $w(v, k-1)$ in descending order. This can be done in linear time [9]. Furthermore, the algorithm maintains a $y$-coordinate $y_L$ which is the $y$-coordinate of the horizontal edge of $L$ between two events.

In the beginning $U$ and $L$ start at the upper and lower end point of $w$. $U$ just follows the upper boundary of $H_w$. Hence, we only have to describe how to compute $L$. So assume the vertical edge $e$ of $H_w$ is the current event point of the plane sweep. At edge $e$ we update $L_{k-1}$, $L$, $U$, and $y_L$. Since $e$ belongs to the upper boundary of $H_w$, we add $e$ to $U$. We remove all the intervals in $L_{k-1}$ whose lower end point is above the lower end point of $e$ since the 2-areas of these intervals do not extend to the right of $e$. Let the removed intervals be $w(v_1, k-1), \ldots, w(v_m, k-1)$. The points in $A_k(w, k-1) \cap H_w$ to the right of $e$ are contained in the 1-areas of $w(v_1, k), \ldots, w(v_m, k)$. Let $y_{max}$ be the highest $y$-coordinate of a lower end point of $w(v_1, k), \ldots, w(v_m, k)$. If $y_{max}$ is greater than $y_L$, then we add a vertical edge at the current position with lower end point at $y_L$ and upper end point at $y_{max}$ to $L$ and set $y_L$ to $y_{max}$. Otherwise, $y_L$ remains unchanged and we continue the current horizontal edge of $L$; see Figure 3a. This ensures that all the points and only those points on the sweep-line between $U$ and $L$ either reach the interval $w(v, k)$ with one link or the interval $w(v, k-1)$ with 2 links, for all vertices $v$ in $V_{k-1}(w)$. The algorithm stops if either the end of $H_w$ is reached or $y_{max}$ goes above $U$. The correctness and linearity of the algorithm are straightforward.

3.1.4 Computing $A_k(w, k-2)$

Finally, we consider the case $d = k - 2$. By Lemma 3.3 the topmost intervals of $w$ play an important role. So let $I_1$ be the topmost $(k - 2)$-interval on $w$
and $v_1$ be the corresponding vertex in $V_{k-2}(w)$, i.e., $I_1 = w(v_1, k - 2)$. The $(k-1)$-interval of $v_1$ is denoted by $J_1$.

Similarly let $J_2$ be the topmost $(k-1)$-interval and $v_2 \in V_{k-2}(w)$ the vertex with $J_2 = w(v_2, k - 1)$. The $(k - 2)$-interval of $v_2$ is denoted by $I_2$.

**Lemma 3.4** If $v_1$ and $v_2$ are defined as above, then $A_k(w, k - 2)$ is the intersection of $A_k(v_1)$ and $A_k(v_2)$.

**Proof:** Since $A_k(w, k - 2)$ is the intersection of all regions $A_k(v)$, with $v \in V_{k-2}(w)$, the inclusion $A_k(w, k - 2) \subseteq A_k(v_1) \cap A_k(v_2)$ immediately follows. Hence, it remains to prove the opposite inclusion. Therefore, let $p$ be a point in the intersection of $A_k(v_1)$ and $A_k(v_2)$. $A_k(v)$ is the union of the 3-area of $w(v, k - 2)$ and the 2-area of $w(v, k - 1)$ by Lemma 3.2. If $p$ is in the 3-area of $I_1$, then $p$ is in the 3-area of all $(k-2)$-intervals by Lemma 3.3 and, therefore, in $A_k(w, k - 2)$. The same holds for the 2-area of $J_2$ and all $(k-1)$-intervals.

So assume that $p$ is contained in $A_k(v_1) \cap A_k(v_2)$ but neither in the 3-area of $I_1$ nor in the 2-area of $J_2$. Hence, $p$ is a point in the intersection of the 2-area of $J_1$ and the 3-area of $J_2$. Let $h$ be the maximal horizontal chord in $H$ with left end point on the bottommost point $p_2$ of $I_2$. If $p$ is above $h$, then $p$ is in the 2-area of $J_2$ which contradicts our choice of $p$. If $p$ is below $h$, then there is a path of three links from $p$ to $p_2$ with the last link collinear with $h$. Hence, the second to last link $l$ is vertical and below $h$. Since $p$ is in
a histogram, $l$ can be moved to $w$ without encountering a boundary point of $H$. Therefore, $p$ is in the 3-area of $I_1$; see Figure 3b. Again we have a contradiction.

Since it is straightforward to compute the 2- and 3-area of an interval in time linear in $H$ as well as their unions and intersections — because all of these sets are horizontally monotone — $A_k(w, k - 2)$ can be computed in time $O(|H| + |P_w|)$.

3.2 Computing $A_k(H)$

Now that we know how to compute $A_k(w)$ for each window $w$, we still have to show how to compute the intersection over all $A_k(w)$ efficiently. Instead of computing the intersection $\bigcap_{w \in H} A_k(w)$ directly, we again make use of Corollary 3.1 and consider the intersection of the regions $A_k(w, d)$ with $w \in H$ and $k - 2 \leq d \leq k$. We divide the regions $A_k(w, d)$ into nine classes depending on whether $w$ is an east window, a west window, or the base and on the value of $d$. We then compute the intersection in each of the classes which leaves us with nine horizontally monotone regions that then in turn are intersected. Since the intersection of horizontally monotone sets can be computed in linear time, we will not address the final intersection.

Without loss of generality we only deal with regions of west windows and the base of $H$. So first consider the regions of west windows with $d = k$. In this case $A_k(w, k)$ is contained in the 1-area of $w$. Since the 1-areas of two west windows are disjoint, there can be only one region with $d = k$ unless $A_k(H)$ is empty.

3.2.1 Relevant regions

So we are left with the regions of west windows for $d = k - 1$ and $d = k - 2$. Before we consider their intersection we need the concept of relevant regions. We say a region $A_k(w, d)$ is irrelevant if there is some $A_k(w', d)$ in the same class with $A_k(w', d) \subseteq A_k(w, d)$ since we are only interested in the intersection of the regions $A_k(w, d)$. A region that is not irrelevant is called relevant. To characterize the regions $A_k(w, d)$ that are relevant we need the following two observations, the first of which we state without proof.

Lemma 3.5 Let $\delta \geq 2$. If $I_1$ and $I_2$ are two intervals on different west edges such that $I_1$ is contained in the 2-area of $I_2$, then the $\delta$-area of $I_1$ is contained in the $\delta$-area of $I_2$.
For each window \( w \), we denote a topmost interval \( w(v, d) \) with \( v \in \mathcal{V}_d(w) \) by \( I_d(w) \).

**Lemma 3.6** If \( A_k(w, d) \) and \( A_k(w', d) \) are two regions in the same class with \( k - 2 \leq d \leq k - 1 \) and the 2-area of \( I_d(w) \) contains \( w' \), then \( A_k(w, d) \) contains \( A_k(w', d) \).

**Proof:** First we assume that \( d = k - 1 \). In this case \( A_k(w', k - 1) \) is contained in the 2-area of \( w' \) which in turn is contained in the 2-area of \( I_{k-1}(w) \) by Lemma 3.5. Lemma 3.3 implies that 2-area of \( I_{k-1}(w) \) is contained in \( A_k(w, k - 1) \) which completes the proof for the case \( d = k - 1 \).

In the case \( d = k - 2 \) a similar argument with 3-areas holds. \( \Box \)

With the above lemma it is easy to see that all relevant regions for one class can be found in linear time, simply by determining the windows that are contained in the 2-areas of the intervals \( I_{k-1}(w) \) and \( I_{k-2}(w) \). This can be done by a left to right scan of the boundary of \( H \) for the set of west windows and a right to left scan of the boundary of \( H \) for the set of east windows.

### 3.2.2 Computing the intersection of the regions \( A_k(w, k - 1) \)

First we turn to the case \( d = k - 1 \). There can be only two relevant regions of west windows with \( d = k - 1 \).

**Lemma 3.7** If \( A_k(w, k - 1) \), \( A_k(w', k - 1) \), and \( A_k(w'', k - 1) \) are three relevant regions, then \( A_k(w, k - 1) \cap A_k(w', k - 1) \cap A_k(w'', k - 1) = \emptyset \).

**Proof:** First recall that \( a_2(I_{k-1}(w)) \) is the 2-area of \( I_{k-1}(w) \) and that \( a_1(w) \) is the 1-area of \( w \) in \( H \).

We have \( A_k(w, k - 1) \subseteq a_2(I_{k-1}(w)) \cup a_1(w) \) by Corollary 3.1 and Lemma 3.2, and hence, if suffices to show that

\[
\emptyset = (a_2(I_{k-1}(w)) \cup a_1(w)) \cap (a_2(I_{k-1}(w')) \cup a_1(w')) \cap (a_2(I_{k-1}(w'')) \cup a_1(w'')).
\]

Note that if we can prove that any pair of 1-areas has an empty intersection and similarly that any pair of 2-areas has an empty intersection, then we are done. This follows if the above equation is rewritten in disjunctive normal form since the new formula will consist of eight terms, each being
the intersection of three areas. So each term is the intersection of at least two 1-areas or two 2-areas.

Clearly, the intersection of two 1-areas is empty since all the windows are west windows.

We prove that a pair of 2-areas has an empty intersection by first noting that a 2-area is a histogram with a segment of the base of $\mathbf{H}$ as its base. Without loss of generality we take the areas corresponding to $I_{k-1}(w)$ and $I_{k-1}(w')$ and assume that the window $w$ is to the left of window $w'$. Project a point of $w'$ onto the base. Since $w'$ is not in the 2-area of window $I_{k-1}(w)$ the projected point on the base cannot be there either. But, this is the leftmost point of the 2-area of $I_{k-1}(w')$ so the two 2-areas are disjoint; see Figure 4a.

This completes the case $d = k - 1$.

3.2.3 Computing the intersection of the regions $A_k(w, k - 2)$

The following lemma gives a simple characterization of the intersection of all relevant regions with $d = k - 2$.

Lemma 3.8 If $A_k(w_1, k - 2), \ldots, A_k(w_\ell, k - 2)$ are $\ell$ relevant regions, $\ell \geq 2$, of west edges of $\mathbf{H}$ in order from left to right and $e$ is a horizontal edge of $\mathbf{H}$ with minimum $y$-coordinate on the boundary of $\mathbf{H}$ such that $w_1$ lies to the left of $e$ and $w_\ell$ lies to the right of it, then $\bigcap_{1 \leq i \leq \ell} A_k(w_i, k - 2)$ equals the 2-area of $e$. 

Figure 4: Computing the intersection of the relevant regions $A_k(w_i, k - 1)$ and $A_k(w_i, k - 2)$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{Computing the intersection of the relevant regions $A_k(w_i, k - 1)$ and $A_k(w_i, k - 2)$.}
\end{figure}
PROOF: Take a point $p$ in the 2-area of $e$. Extend a horizontal line segment $h_e$ through $e$ until it reaches both a left and right boundary point. If any window of a region in the class lies below the line segment, it can only be $w_1$ since $e$ is a lowest edge after $w_1$. Hence, the line segment $h_e$ can be reached by an admissible path of two links from any point on the windows $w_1, \ldots, w_\ell$ which implies that $p$ is contained in the intersection of all 3-areas of intervals $I_{k-2}(w_i)$ of vertices with distance $k - 2$ to $w_i$. Hence, $a_2(e)$ is a subset of $A_k(w_i, k - 2)$ by Lemma 3.2.

Now take a point in the intersection $p \in \bigcap_{1 \leq i \leq \ell} A_k(w_i, k - 2)$. Let $h_e$ be defined as before. If $p$ lies above $h_e$ and, say, is to the left of $e$, then there is a window $w_j$ to the right of $e$ such that $p$ is not in the 3-area of that window, thus $p$ cannot reach $w_j$ with three links contradicting $p \in \bigcap_{1 \leq i \leq \ell} A_k(w_i, k - 2)$. Similarly if $p$ lies to the right of $e$, there is some window to the left of $e$ producing the same contradiction.

So $p$ is below or on $h_e$. We can show as in the proof of Lemma 3.4 that $p$ belongs to the intersection of the 3-areas of $I_{k-2}(w_i)$, $1 \leq i \leq \ell$. Hence, the last link to each window $w_i$ is horizontal and the first link from $p$ is also horizontal. It follows that any horizontal edge between the first and the last window can be reached with two links from $p$, in particular $e$; see Figure 4b.

\[\square\]

The pseudo code for how to compute the intersection of the regions associated to windows in some set $S$ of horizontal windows, i.e., the west windows, is shown in Algorithm Handle-Vertical-Windows.

### 3.3 The Region for the Base Window

In this section we show how to compute the region $A_k(c)$. From Lemma 3.1 we know that $A_k(c) = A_k(c, d_e) \cap A_k(c, d_e - 1)$. However, we also know that if $k < d_e$, then $A_k(c)$ is empty and if $k \geq d_e + 2$, then $A_k(c) = H$. Hence, the only non-trivial regions are $A_k(c, k)$, $A_k(c, k - 1)$, and $A_k(c, k - 2)$. Now it is easy to show that $A_k(c, k - 2) = H$ since any point on $c$ can be reached from a point in $H$ with three links. The region $A_k(c, k)$ is a 1-area, i.e., a maximal histogram in $H$, having the intersection of all the intervals $c(v, k)$, with $v$ in $P_c$, as base, since this is the only region that can reach all the intervals $c(v, k)$ with one link. Hence, we only have to concern ourselves with how to compute the region $A_k(c, k - 1)$, this being the difficult part. We show how to do this in the next section.
Algorithm Handle-Vertical-Windows

Input: A maximal histogram $H$ in $P$ with a chord $c$ as base, a set $S$ of vertical windows of $H$, and an integer $k$

Output: The region $\text{Area}_S$ associated to the windows of $S$

1. if there is a window $w \in S$ such that $d_w > k$
   then return $\text{Area}_S := \emptyset$ and exit the algorithm
   /* The set of points in $A_k(H)$ is empty */
   endif

2. Let zero be the subset of windows $w \in S$ for which $d_w = k$
   Let one be the subset of windows $w \in S$ for which $d_w = k$ or $d_w = k - 1$
   Let two be the subset of windows $w \in S$ for which $d_w = k - 1$ or $d_w = k - 2$

3. case [zero] of
   0: $\text{Area}_S(0) := H$
   1: $\text{Area}_S(0) := A_k(w, k)$ with $w$ in zero
   otherwise $\text{Area}_S(0) := \emptyset$
   endcase

4. Remove the irrelevant windows from the set one
   case [one] of
   0: $\text{Area}_S(1) := H$
   1: $\text{Area}_S(1) := A_k(w, k - 1)$ with $w$ in one
   2: $\text{Area}_S(1) := A_k(w, k - 1) \cap A_k(w', k - 1)$ with $w$ and $w'$ in one
   otherwise $\text{Area}_S(1) := \emptyset$
   endcase

5. Remove the irrelevant windows from the set two
   case [two] of
   0: $\text{Area}_S(2) := H$
   1: $\text{Area}_S(2) := A_k(w, k - 2)$ with $w$ in two
   otherwise $\text{Area}_S(2) := a_j(e)$ where $e$ is the horizontal boundary segment of $H$ with lowest $y$-coordinate such that there are windows of two both to the left and right of $e$
   endcase

6. return $\text{Area}_S := \text{Area}_S(0) \cap \text{Area}_S(1) \cap \text{Area}_S(2)$

End Handle-Vertical-Windows
3.3.1 Computing the Region $A_k(c, k - 1)$

From de Berg [2] we know that the intervals $c(v, k - 1)$ and $c(v, k)$, for $v \in V_c(k - 1)$, have one endpoint in common (and, of course, that $c(v, k)$ extends at least as far as $c(v, k - 1)$). So we partition the vertices in $V_c(k - 1)$ into two sets, $V_L$ and $V_R$. The set $V_R$ contains the vertices for which $c(v, k)$ extends to the right of $c(v, k - 1)$ and the set $V_L$ contains the remaining vertices.

We establish properties of the regions $A_k(v)$ corresponding to the vertices in the two sets separately. The two cases are symmetric so we only show the properties for the set $V_L$. We denote the area $\bigcap_{v \in V_L} A_k(v)$ by $L$. Similarly, we define the area $R$ for the vertices in $V_R$. Thus we get $A_k(c, k - 1) = L \cap R$. We will only show how to compute the region $L$ since the region $R$ is symmetric to $L$ and can be computed using a similar algorithm.

For each vertex $v \in V_L$ we denote the end points of the corresponding intervals $c(v, k)$ and $c(v, k - 1)$ by $\alpha_v, \beta_v$, and $\gamma_v$ in such a way that $c(v, k) = [\alpha_v, \gamma_v]$ and $c(v, k - 1) = [\beta_v, \gamma_v]$. We view these points as three designated points ordered from left to right. Hence, it is possible to have $\alpha_v = \beta_v$ but w.l.o.g. we can still view $\alpha_v$ as being to the left of $\beta_v$ in this case.

Let $v_1$ be a vertex in $V_L$ such that $\gamma_{v_1}$ is the leftmost among the set of points $\gamma_v$, with $v$ in $V_L$. It turns out that only a subset of the vertices in $V_L$ is needed to compute $L$. We show this in the following lemma.

**Lemma 3.9** Let $v$ and $v'$ be two vertices, with $v, v' \in V_L$ and such that $\alpha_{v'}$ is to the left of $\alpha_v$. If $\beta_{v'}$ is to the left of $\beta_v$, then

$$A_k(v) \cap A_k(v_1) \subseteq A_k(v').$$

**Proof:** Take a point $p \in A_k(v) \cap A_k(v_1)$. We will show that from $p$ it is possible to reach $v'$ using at most $k$ links.

Suppose first that $p$ lies to the left of $\alpha_{v'}$. This means that $p$ cannot be in $A_1(c(v, k))$ since $\alpha_{v'}$ is to the left of $\alpha_v$; see Figure 5. Then, $p \in A_2(c(v, k - 1))$ and by the histogram property of $H$, some point to the right of $\beta_v$ can be reached with two links since a point to the right of $\beta_v$ can be reached with two links. Hence, $p \in A_2(c(v', k - 1)) \subseteq A_k(v')$.

If $p$ lies between $\alpha_{v'}$ and $\gamma_{v'}$, then one link straight down hits the interval $c(v', k)$ and $p \in A_1(c(v', k)) \subseteq A_k(v')$.

Finally, if the point $p$ lies to the right of $\gamma_{v'}$, then $p$ is not in $A_1(c(v_1, k))$ since $\gamma_{v_1}$ is to the left of $\gamma_{v'}$. Hence, $p \in A_2(c(v, k - 1))$ and by the histogram
property of \( \mathbf{H} \), some point to the left of \( \gamma_{v'} \) can be reached with two links since a point to the left of \( \gamma_{v_k} \) can be reached with two links, i.e., \( p \in a_2(c(v', k - 1)) \subseteq A_k(v') \); see Figure 5.

Let \( \mathcal{V}'_{L} \) be the subset of \( \mathcal{V}_L \) containing the vertices \( v \) for which there is no vertex \( v^* \) with \( \alpha_v \) to the left of \( \alpha_{v'} \) and \( \beta_v \) to the left of \( \beta_{v'} \). By the previous lemma it is clear that \( L = \bigcap_{v \in \mathcal{V}'_L} A_k(v) \cap A_k(v_i) \).

Computing the set \( \mathcal{V}'_L \) is done in the following way. Keep a list of the triples \( (\alpha_v, \beta_v, \gamma_v), v \in \mathcal{V}_L \). From [9, Section 3.2] we can assume that the list is sorted on the points \( \beta_v \) from right to left. Now, scan the list from beginning to end and look at the adjacent pairs of elements. If \( \alpha_{v_{i+1}} \) is to the left of \( \alpha_{v_i} \) then remove the element \( (\alpha_{v_{i+1}}, \beta_{v_{i+1}}, \gamma_{v_{i+1}}) \) from the list.

From Lemma 3.9 we have, if we denote \( \bigcap_{v \in \mathcal{V}'_L} A_k(v) \) by \( L' \), that \( L = L' \cap A_k(v_i) \).

Let \( H_L \) be the part of \( H \) to the left of the rightmost point \( \alpha_{v'} \), with \( v \in \mathcal{V}'_L \) and let \( H_m \) be the part of \( H \) to the right of \( H_L \) and to the left of \( \gamma_{v_k} \), with \( v \in \mathcal{V}'_L \). Let \( H_r \) be \( H - H_L - H_m \). Similarly define the parts \( L'_L, L'_m \), and \( L'_r \) as the intersections of \( L' \) with the corresponding part of \( H \). It is easy to see that \( H_m = L'_m \), since any point in \( H_m \) reach all the intervals \( c(v, k) \), where \( v \in \mathcal{V}'_L \), with one downward link.

The lower boundary of \( L'_L \) is the base of \( H_L \) and we show with a plane sweep algorithm how to compute the upper boundary of \( L'_L \); see Figure 6(a).

Assume that \( \mathcal{V}'_L \) contains \( j \) vertices and that the points \( \beta_{v_1}, \ldots, \beta_{v_j} \), with \( v_i \in \mathcal{V}'_L \), are sorted from left to right. By Lemma 3.9 we know that, for all vertices in \( \mathcal{V}'_L \), the interval \([\alpha_{v_i}, \beta_{v_i}]\) contains the interval \([\alpha_{v_{i-1}}, \beta_{v_{i-1}}]\). The plane sweep is performed simultaneously towards the left and right starting
Algorithm Handle-Base
Input: A maximal histogram $H$ in $P$ with a chord $c$ as base and an integer $k$
Output: A region $A_{\text{base}}$ associated to the base $c$ of $H$

\begin{algorithmic}
  \State \textbf{if} $k - d_c < 0$
  \State \hspace{1em} \textbf{then return} $A_{\text{base}} := \emptyset$
  \State \textbf{elseif} $k - d_c = 0$
  \State \hspace{1em} \textbf{then return} $A_{\text{base}} := A_k(c, k) \cap A_k(c, k - 1)$
  \State \textbf{elseif} $k - d_c = 1$
  \State \hspace{1em} \textbf{then return} $A_{\text{base}} := A_k(c, k - 1)$
  \State \textbf{elseif} $k - d_c \geq 2$
  \State \hspace{1em} \textbf{then return} $A_{\text{base}} := H$

  \end{algorithmic}

end if

/* The region $A_k(c, k)$ is the maximal histogram having the intersection of all the intervals $c(v, k)$, with $v \in P_c$, as base. The region $A_k(c, k - 1) = L \cap R$ where $L = L' \cap A_k(v_l)$ and $L'$ is the intersection $\bigcap_{v \in V_{L_k}'} A_k(v)$ and $v_l$ is the vertex in $V_{L_k}$ with the leftmost right end point of the interval $c(v_l, k)$. The region $R$ is defined and computed symmetrically. */

End Handle-Base

from the innermost interval $[\alpha_v, \beta_v]$. The objective of the $i$th step of the sweep is to find the $y$-coordinate of the lowest point on the upper boundary of $H_i$ above the interval $[\alpha_v, \beta_v]$. We scan the boundary of $H_m$ between the points $\alpha_v$ and $\beta_v$ for the lowest boundary point $p_l$. The $y$-coordinate of $p_l$ together with the $x$-coordinate of $\alpha_v$ defines a point $b_1$. Now, assuming that we have computed $b_i$, we can find the point $b_{i+1}$ as follows. Scan the part of the boundary between $\alpha_v$ and $\alpha_{v+i}$ in $H_i$ as well as the part between $\beta_v$ and $\beta_{v+i}$ in $H_m$ for the lowest point. Take the minimum $y$-coordinate of this point and $b_i$ and the $x$-coordinate of $\alpha_{v+i}$ to get the point $b_{i+1}$. The upper boundary of $L'_i$ is the uppermost $xy$-monotone chain inside $H$ that passes through the points $b_i$, with $v_i \in V_{L'_i}$, and $b_{i+1}$. To prove this consider a point $p$ below the chain. The point $p$ can reach the intervals $c(v, k)$ with one link for the vertices in $V_{L}'$ for which $\alpha_v$ lies to the left of $p$. Thus, we must show that for the remaining vertices in $V_{L}'$, the intervals $c(v, k - 1)$ can be reached with two links. But this follows trivially since the computed chain ensures that if $p$ lies below it, it is possible to reach the $c(v, k - 1)$-intervals for the remaining vertices using a link towards the right and a down link. The same argument applies to show that if $p$ lies above the chain, then there is some
vertex $v$ in $V_L^k$ for which the $c(v, k)$ interval can not be reached with one link and the $c(v, k - 1)$ interval can not be reached with two links.

Finally, the region $L'_m$ is just the set of points that can reach some point on the base of $H_m$ with two links; see Figure 6(b).

It is easy to see that the intersection of $L'$ and $A_k(v)$ can be computed in linear time since both regions are monotone with respect to the base $c$.

The pseudo code showing how to compute the region associated with the base is depicted in Algorithm Handle-Base.

To see that our algorithm uses linear time in total note that $A_k(H)$ is the intersection of a constant number of possibly empty regions in $H$. Each region can be computed in linear time, which is what we have shown in this section, and since the regions are all monotone w.r.t. the base of $H$ their intersection can be computed in linear time. Thus, we have proven the main result of this section.

**Theorem 1** Let $P$ be a simple rectilinear polygon and $H$ a maximal histogram in $P$ having the rectilinear chord $c$ in $P$ as its base. The algorithm described above computes the region inside $H$ from which all other points in $P$ can be reached with at most $k$ links. The algorithm runs in linear time in the size of $P$.
Algorithm Part-in-Histogram

Input: A simple rectilinear polygon $\mathcal{P}$, a maximal histogram $\mathcal{H}$ in $\mathcal{P}$ with a chord $c$ as base, and an integer $k$

Output: The set $A_k(\mathcal{H})$ of points in $\mathcal{H}$ from which all other points in $\mathcal{P}$ can be reached with at most $k$ links

1 Identify the windows of $\mathcal{H}$ and for each window compute $d_w$, the distance from the window $w$ to the furthest point in $\mathcal{P}_w$

2 if $k - d_w > 2$, for all the windows in $\mathcal{H}$
   then return $A_k(\mathcal{H}) := \mathcal{H}$
   /* The set of points in $A_k(\mathcal{H})$ is the complete histogram */
   endif

3 Let $\mathcal{W}$ be the set of west windows and let $\mathcal{E}$ be the set of east windows

4 Apply the subroutine $\text{Handle-Vertical-Windows}$ with $\mathcal{W}$ as parameter to compute the area $\text{Area}_W$ associated to the west windows

5 Apply the subroutine $\text{Handle-Vertical-Windows}$ with $\mathcal{E}$ as parameter to compute the area $\text{Area}_E$ associated to the east windows

6 Apply the subroutine $\text{Handle-Base}$ to compute the area $\text{Area}_\text{base}$ associated to the base

7 /* The result is the intersection of all the regions */
   return $A_k(\mathcal{H}) := \text{Area}_W \cap \text{Area}_E \cap \text{Area}_\text{base}$

End Part-in-Histogram

We show the pseudo code for the main routine in Algorithm Part-in-Histogram. It calls $\text{Handle-Vertical-Windows}$ and $\text{Handle-Base}$ and then computes the intersection of the returned regions to produce the set of points in $\mathcal{H}$ that can reach all vertices in $\mathcal{P}$ using at most $k$ links.

Together with the fact that we can compute the rectilinear link diameter in linear time [9] and the result in Lemma 2.1 we have the following corollary to Theorem 1.

Corollary 3.2 The region $\text{lc}(\mathcal{P}) \cap \mathcal{H}$ can be computed in linear time.

4 Finding a Central Chord

In this section we are concerned with computing a central chord. A chord $c$ is a central chord of $\mathcal{P}$ if $c$ splits $\mathcal{P}$ into two subpolygons $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $\varepsilon(c, \mathcal{P}_i) \geq R(\mathcal{P}) - 1$, for $i = 1$ and 2. Recall that $\varepsilon(c, \mathcal{P}_i)$ is the maximum link distance between $c$ and any point in $\mathcal{P}_i$. If a central chord exists, then
it is close to the link center. This can be seen as follows. Let $c$ be a central chord in $P$ that splits $P$ into the two subpolygons $P_1$ and $P_2$. If $p$ is a point in $lc(P) \cap P_1$ and $v$ is a vertex in $P_2$ with $d(v, c) = \varepsilon(c, P_2)$, then $d(p, c) \leq d(p, v) - d(v, c) + 1 \leq R(P) - \varepsilon(c, P_2) + 1 \leq 2$ by Corollary 2.1. Hence, if $c$ is a central chord, then $d(p, c) \leq 2$, for all points $p \in lc(P)$. Computing a central chord is a main step in our algorithm to compute the link center.

In order to show the existence of a central chord and to give an algorithm for its computation the following lemma is crucial.

**Lemma 4.1** If $P$ is a simple rectilinear polygon with diameter $D$, then there is a chord $c$ that splits $P$ into two subpolygons $P_1$ and $P_2$ with $|D/2| \leq \varepsilon(c, P_i) \leq [D/2] + 1$, for $i = 1, 2$.

**Proof:** Let $v_1$ and $v_2$ be two vertices in $P$ with $d(v_1, v_2) = D$, $\mathcal{P}$ a shortest path from $v_1$ to $v_2$, and $k = [D/2]$. We denote the $(k + 1)$-st link of $\mathcal{P}$ by $l_{k+1}$ and the maximal line segment containing $l_{k+1}$ by $s_{k+1}$. Let $I_1$ be the interval on $s_{k+1}$ that $v_1$ reaches with $k$ links and $I_2$ the interval on $s_{k+1}$ that $v_2$ reaches with $D - k - 1$ links. Clearly, $I_1$ and $I_2$ do not intersect.

Without loss of generality we assume that $s_{k+1}$ is vertical, that $v_1$ is contained in the subpolygon that is to the left of $s_{k+1}$, and that $I_1$ is above $I_2$. Let $s$ be the part of $s_{k+1}$ between $I_1$ and $I_2$.

First suppose that $D = 2k$. Let $H_s$ be the maximal histogram with base $s$ to the left of $s$, and $e_s$ the closest vertical edge of $H_s$ to $s$. Furthermore, let $e$ be the vertical edge of $P$ containing $e_s$ and $c$ the vertical chord above $e$. The chord $c$ splits $P$ into the subpolygons $P_l$ to the left of $c$ and $P_r$ to the right of $c$. We show that $c$ satisfies $k \leq \varepsilon(c, P_i) \leq k + 1$, for $i = l, r$.

Note that $v_1$ is in $P_l$ and $v_2$ is in $P_r$. For illustration refer to Figure 7a. We assume for now that $d(v_1, c) \geq k$ and $d(v_2, c) \geq k$. Hence, if $v$ is a vertex in $P_l$, then $d(v, c) \leq d(v, v_1) - d(c, v_1) + 1 \leq D - k + 1 = k + 1$ by Corollary 2.1. Similarly, if $v$ is a vertex in $P_r$, then $d(v, c) \leq k + 1$ and $k \leq \varepsilon(c, P_i) \leq k + 1$, for $i = l, r$ as claimed.

So it remains to be shown that $d(v_1, c) \geq k$ and $d(v_2, c) \geq k$. Since $v_2$ can reach the maximal vertical line segment $s_e$ that is collinear with $e$ with $k - 1$ links, we have $d(v_1, c) \geq d(v_1, s_e) \geq k$, for otherwise $d(v_1, v_2) \leq d(v_1, s_e) + d(s_e, v_2) + 1 \leq k - 1 + k - 1 + 1 < D$. To see that $d(v_2, c) \geq k$ note that $c$ is above $I_2$; this implies that $v_2$ cannot reach $c$ with $k - 1$ links and $d(v_2, c) \geq k$. Hence, $k + 1 \geq \varepsilon(c, P_l) \geq k$ and $k + 1 \geq \varepsilon(c, P_r) \geq k$ which proves the lemma for even $D$. 

21
Now suppose that $D = 2k + 1$. Let $q_o$ be the middle point of $s$ and $c_s$ the horizontal chord through $q_o$. The chord $c_s$ also splits $P$ into two subpolygons. Let $P_a$ be the subpolygon above $c_s$ and $P_b$ the subpolygon below $c_s$. It is our second candidate for a chord $c$ with $k \leq \varepsilon(c, P_i) \leq k+1$. Note that $v_1$ belongs to $P_a$ and $v_2$ belongs to $P_b$.

The distance from $v_1$ and $v_2$ to $c_s$ is at least $k+1$ and $k$, respectively, which can be seen as follows. The chord $c_s$ is contained in $P_r$ and does not intersect $c$ by our choice of $q_o$. Since a shortest path from $v_1$ to a point in $P_r$ crosses $c$, we have $d(v_1, c_s) \geq d(v_1, c) + 1 = k + 1$. Furthermore, since $c_s$ intersects $s$, $d(v_2, c_s) \geq d(v_2, s) - 1 = k$.

Since $d(v, v_1) \leq D$, for all vertices $v$ in $P_b$, and $d(v_1, c_s) \geq k + 1$, there is no vertex $v$ in $P_b$ with $d(v, c) \geq k + 1$ by Corollary 2.1. Similarly, since $d(v, v_2) \leq D$, for all vertices $v$ in $P_a$, and $d(v_1, c_s) \geq k$, there is no vertex $v$ in $P_a$ with $d(v, c) \geq k + 1$. We obtain $k + 2 \geq \varepsilon(c_s, P_a) \geq k + 1$ and $k + 1 \geq \varepsilon(c_s, P_b) \geq k$.

If $\varepsilon(c_s, P_b) = k + 1$ and $v_b$ is a vertex in $P_b$ with $d(v_b, c_s) = k + 1$, then, since again $d(v, v_b) \leq D$, for all vertices $v$ in $P_a$, $\varepsilon(c_s, P_a) \leq k + 1$ and $c_s$ satisfies the lemma. If $\varepsilon(c_s, P_b) = k$ and $\varepsilon(c_s, P_a) = k + 2$, then $c_s$ also satisfies the lemma.

So assume that $\varepsilon(c_s, P_b) = k$ and $\varepsilon(c_s, P_a) = k + 2$. In this case $c_s$ does not satisfy the lemma but we show in the following that there is another

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure7}
\caption{Illustrating the proof of Lemma 4.1.}
\end{figure}
chord that does.

There is a vertex \( v_a \) in \( P_a \) with distance \( k+2 \) to \( c_s \). Consider the maximal histogram \( H_{c_s} \) above \( c_s \); see Figure 7b. Let \( w \) be the window of \( H_{c_s} \) that cuts off the subpolygon that contains \( v_a \). Since \( d(v_a, c_s) \geq k + 2 \) and \( c_s \) can be reached from \( w \) with one link, we have \( d(v_a, w) \geq k + 1 \). Since \( v_2 \) is below \( c_s \) and \( d(v_2, c_s) \geq k \), we have \( d(v_2, w) \geq k + 1 \) and \( w \) satisfies the lemma. This completes the proof. \( \Box \)

Note that if we construct a chord \( w \) different from \( c \) and \( c_s \) in the above proof, then the eccentricity of \( w \) in both subpolygons is \([D(P)]/2\). We will make use of this observation later. Before we consider central chords, we first prove the second half of Lemma 2.1 with the help of the above lemma.

**Corollary 4.1** If \( P \) is a simple rectilinear polygon, then

\[ R(P) \leq [D(P)/2] + 1. \]

**Proof:** Let \( c \) be a chord in \( P \) that splits \( P \) into two subpolygons \( P_1 \) and \( P_2 \) with \([D(P)/2] + 1 \geq \varepsilon(c, P_i) \geq [D(P)/2] \), for \( i = 1, 2 \). We construct one point \( p \) on \( c \) with \( d(p, v) \leq [D(P)/2] + 1 \), for all vertices \( v \) of \( P \). Let \( k = [D(P)/2] \) and \( v \) be a vertex in \( P \). If \( D(P) = 2k + 1 \), then let \( p \) be some point on \( c \). Since \( d(v, c) \leq k + 1 \), we have \( d(p, c) \leq k + 2 = [D(P)/2] + 1 \).

Now assume that \( D(P) = 2k \) and \( v_1 \) and \( v_2 \) are two vertices in \( P \). If \( v_1 \) and \( v_2 \) are both in \( P_1 \) or both in \( P_2 \), then \( c(v_1, k + 1) \) intersects \( c(v_2, k + 1) \); otherwise, \( d(v_1, v_2) \geq k + 1 + k + 1 - 1 > D(P) \) by Lemma 2.3. If \( v_1 \) is in \( P_1 \) and \( v_2 \) in \( P_2 \), then \( c(v_1, k + 1) \) also intersects \( c(v_2, k + 1) \); otherwise \( d(v_1, v_2) \geq k + 1 + k + 1 > D(P) \) by Lemma 2.2. Hence, \( I = \bigcap_{v \in P} c(v, k + 1) \) is non-empty and a point \( p \) in \( I \) can reach all points with at most \( k + 1 = [D(P)/2] + 1 \) links. \( \Box \)

Using the above results we are now able to show that a central chord always exists.

**Lemma 4.2** If \( P \) is a simple rectilinear polygon, then there exists a central chord.

**Proof:** If \( D(P) = 2k \), then \( R(P) \leq k + 1 \) and the chord constructed in the proof of Lemma 4.1 is already a central chord. So we assume in the following that \( D(P) = 2k + 1 \). Let \( c \) be a chord in \( P \) that splits \( P \) into two subpolygons \( P_1 \) and \( P_2 \) with \( k + 1 \geq \varepsilon(c, P_1) \geq k \). If \( \varepsilon(c, P_1) = \varepsilon(c, P_2) = k + 1 = [D(P)/2] \), then \( c \) is a central chord. If, on the other hand, \( \varepsilon(c, P_1) = \varepsilon(c, P_2) = k = [D(P)/2] \) and \( p \) is a point on \( c \), then
\(d(p, v) \leq k + 1\), for all vertices \(v\) of \(P\) and, hence, \(R(P) = k + 1 = \lceil D(P) / 2 \rceil\) which again implies that \(c\) is central chord.

Therefore, we assume in the following that \(\varepsilon(c, P_1) = k + 1\) and \(\varepsilon(c, P_2) = k\). If the intersection of all the intervals \(c(v, k + 1)\) taken over all vertices in \(P_1\) is non-empty, then again \(R(P) \leq k + 1\) and \(c\) is a central chord. Finally, assume there are two vertices \(v_1\) and \(v_2\) in \(P_1\) with disjoint intervals \(c(v_1, k + 1)\) and \(c(v_2, k + 1)\). Let \(p\) be a point on \(c\) between \(c(v_1, k + 1)\) and \(c(v_2, k + 1)\) and \(\tau\) the chord through \(p\) orthogonal to \(c\). Clearly, the vertices \(v_1\) and \(v_2\) are on different sides of \(\tau\). Since \(p\) can be reached with one link from any point on \(\tau\) and \(p\) does not belong to \(c(v_1, k + 1)\), we have \(d(v_1, \tau) \geq k + 1\). Similarly, we have \(d(v_2, \tau) \geq k + 1\). Since \(R(P) \leq k + 2\), \(\tau\) is a central chord. □

<table>
<thead>
<tr>
<th>Algorithm Central-Chord</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A simple rectilinear polygon (P) and a shortest path (P) between two vertices (v_1) and (v_2) of (P) that span the diameter</td>
</tr>
<tr>
<td><strong>Output:</strong> A central chord of (P)</td>
</tr>
<tr>
<td>1 (k = \lceil D(P) / 2 \rceil); (l_{k+1}) the (k + 1)-st link of (P)</td>
</tr>
<tr>
<td>2 (s) the part of (l_{k+1}) between (l_{k+1}(v_1, k)) and (l_{k+1}(v_2, D(P) - k - 1))</td>
</tr>
<tr>
<td>3 (e) the rightmost edge of the left maximal histogram of (s)</td>
</tr>
<tr>
<td>4 (c) the chord above (e); let (P_l) and (P_r) be the two subpolygons of (c)</td>
</tr>
<tr>
<td>5 <strong>if</strong> (D(P)) is even</td>
</tr>
<tr>
<td>\hspace{1cm} then output (c)</td>
</tr>
<tr>
<td>\hspace{1cm} else (c_s) the horizontal chord through the middle point of (s)</td>
</tr>
<tr>
<td>\hspace{1cm} let (P_s) and (P_h) be the two subpolygons of (c_s)</td>
</tr>
<tr>
<td>\hspace{1cm} if (\varepsilon(c_s, P_a) - \varepsilon(c_s, P_b) = 2)</td>
</tr>
<tr>
<td>\hspace{1cm} then let (H_{c_s}) be the maximal histogram above (c_s)</td>
</tr>
<tr>
<td>\hspace{1cm} let (w) a window of (H_{c_s}) that cuts off a vertex (v_a) with</td>
</tr>
<tr>
<td>\hspace{1cm} (d(v_a, c_s) = k + 2)</td>
</tr>
<tr>
<td>\hspace{1cm} output (w)</td>
</tr>
<tr>
<td>\hspace{1cm} elseif (\varepsilon(c_s, P_a) = \varepsilon(c_s, P_b)) or</td>
</tr>
<tr>
<td>\hspace{1cm} (\varepsilon(c_s, P_a) = k) and (\bigcap_{v \in P_a} c_s(v, k + 1) \neq \emptyset) or</td>
</tr>
<tr>
<td>\hspace{1cm} (\varepsilon(c_s, P_b) = k) and (\bigcap_{v \in P_b} c_s(v, k + 1) \neq \emptyset)</td>
</tr>
<tr>
<td>\hspace{1cm} then output (c_s)</td>
</tr>
<tr>
<td>\hspace{1cm} else \ compute and output (\tau)</td>
</tr>
<tr>
<td><strong>endif</strong></td>
</tr>
<tr>
<td><strong>End Central-Chord</strong></td>
</tr>
</tbody>
</table>

1Note that this case can only occur if the chord constructed is the chord \(c_s\) in the proof of Lemma 4.1.

24
Since the proofs of the above lemmas are constructive, we obtain Algorithm **Central Chord** to construct a central chord of \( P \). Here the orientation of the middle link \( l_{k+1} \) of a shortest path \( P \) between \( v_1 \) and \( v_2 \), as well as other geometric orientations are assumed to be as in the proof of Lemma 4.1.

Since the above algorithm consists of a constant number of chord-, maximal histogram-, eccentricity-, interval-, and interval intersection-computations, it runs in linear time and we obtain the following theorem.

**Theorem 2** Let \( P \) be a simple rectilinear polygon. A central chord of \( P \) can be computed in linear time.

## 5 Computing the Link Center

We now turn to the computation of the link center of \( P \). More precisely, we show the following result.

**Theorem 3** If \( c \) is a central chord of \( P \) that splits \( P \) into two parts \( P_1 \) and \( P_2 \), then \( \text{lc}(P) \cap P_1 \) and \( \text{lc}(P) \cap P_2 \) can be computed in linear time.

Throughout this section we assume that we are given a central chord \( c \) that splits \( P \) into two parts \( P_1 \) and \( P_2 \). We denote the eccentricity of \( c \) in \( P_i \) by \( \varepsilon_i \), for \( i = 1, 2 \), and assume that \( c \) is vertical and that \( P_1 \) is to the left of \( c \) and \( P_2 \) to the right. We only show how to compute the part of the link center in \( P_2 \).

As we have pointed out before, the part of the link center in \( P_2 \) is contained in the region that can be reached from \( c \) with \( R - \varepsilon_1 + 1 \) links. Since \( c \) is a central chord, we have \( \varepsilon_1 \geq R - 1 \). We distinguish two cases. If \( \varepsilon_1 \geq R \), then \( R - \varepsilon_1 + 1 \leq 1 \) and \( \text{lc}(P) \cap P_2 \) is contained in the maximal histogram \( H_c \) of \( c \) in \( P_2 \). Since we know how to compute the link center of a polygon inside a histogram in linear time by Corollary 3.2, we assume from now on that \( \varepsilon_1 = R - 1 \). By Corollary 2.1 the link center is then contained in the 2-area of \( c \) in \( P_2 \). In the following we show how to reduce this case to the computation of the link center in a number of histograms. In order to do so let \( I_1 \) be the set of intervals \( \{c(v, \varepsilon_1) \mid v \in P_1 \} \) and \( I_1 \) the intersection of all intervals in \( I_1 \), i.e., \( I_1 = \bigcap I_1 \). Recall that \( I_1 \) can be computed in linear time [2].

First, suppose that \( I_1 = \emptyset \). Let \( \overline{T}_1 \) be the smallest interval that intersects all the intervals in \( I_1 \); see Figure 8. Let \( q_m \) be the middle point of \( \overline{T}_1 \) and \( h \) be the horizontal line segment through \( q_m \) towards the right. Note that \( h \)
is a chord in \( P_2 \). Let \( H_a \) be the maximal histogram in \( P_2 \) above \( h \) and \( H_b \) be the maximal histogram below \( h \).

**Lemma 5.1** If \( \varepsilon_1 = R - 1 \) and \( I_1 = \emptyset \), then \( lc(P) \cap P_2 \subseteq H_a \cup H_b \cup H_c \).

**Proof:** Let \( p \) be a point in \( lc(P) \cap P_2 \), say in the subpolygon of \( P_2 \) above \( h \), but not in \( H_a \cup H_b \); see Figure 8. We show that \( p \) is contained in \( H_c \).

Since \( I_1 \) is empty, there is a a vertex \( v \) in \( P_1 \) with \( d(v, c) = \varepsilon_1 \) such that \( I = c(v, \varepsilon_1) \) is below \( h \) and, hence, below \( p \). Let \( \mathcal{P} \) be a shortest path from \( v \) to \( p \). Since \( p \) is not in \( H_a \cup H_b \), we have \( d(p, I) \geq 3 \); therefore, \( \mathcal{P} \) does not intersect \( I \) and at least \( \varepsilon_1 + 1 \) links of \( \mathcal{P} \) are needed to reach \( c \). This implies that \( p \) is contained in \( H_c \) since \( \lambda(\mathcal{P}) \leq \varepsilon_1 + 1 = R \).

The above lemma implies that \( lc(P) \cap P_2 \) can be computed with three applications of the algorithm to compute the link center inside a histogram. By Corollary 3.2 this takes only linear time.

So we are left with the case that \( \varepsilon_1 = R - 1 \) and \( I_1 \) non-empty. Recall that the 2-area of \( I_1 \) in \( P_2 \) is denoted by \( a_2(I_1, P_2) \).

**Lemma 5.2** If \( \varepsilon_1 = R - 1 \) and \( I_1 \neq \emptyset \), then \( lc(P) \cap P_2 \subseteq a_2(I_1, P_2) \cup H_c \).

**Proof:** The proof is analogous to the proof of Lemma 5.1. \( \square \)

Hence, we obtain the following algorithm for the computation of the link center of \( P \).
**Algorithm** Link-Center

**Input:** A simple rectilinear polygon $\mathbf{P}$ and a central chord $c$

**Output:** The link center of $\mathbf{P}$ to one side of $c$

1. For $i = 1, 2$, let $\varepsilon_i := \varepsilon(c, \mathbf{P}_i)$
2. Compute $H_c$ and apply the subroutine *Part-in-Histogram* to compute the part of $lc(\mathbf{P})$ in $H_c$
3. If $\varepsilon_1 < R$
   - then $I_1 := \bigcap_{v \in \mathbf{P}_1} c(v, \varepsilon_1)$
   - if $I_1 = \emptyset$
     - then Compute $H_a$, $H_b$ and apply the subroutine *Part-in-Histogram* to compute the part of $lc(\mathbf{P})$ in $H_a$ and $H_b$
   - else Apply the subroutine *Link-Center-in-a$_2$(I$_1$, P$_2$)* to compute the part of $lc(\mathbf{P})$ in a$_2$(I$_1$, P$_2$)

End *Link-Center*

5.1 Computing the Link Center in the 2-Area of I$_1$

Our algorithm to compute the part of the link center in the 2-area of I$_1$ is based on the following observation.

**Lemma 5.3** If I$_1$ is defined as above, then I$_1$ can be split into two subintervals I$_{11}$ and I$_{12}$ with \( \varepsilon(I_{1i}, \mathbf{P}_2) = R \), for $i = 1, 2$.

**Proof:** If there is one vertex $v$ in $\mathbf{P}_2$ such that $c(v, R - 1)$ is disjoint from $I_1$, then $d(v, I_1) = R$ and $\varepsilon(I_{1i}, \mathbf{P}_2) = R$; hence, we can choose $I_{11} = I_1$ and $I_{12} = \emptyset$.

If $c(v, R - 1)$ intersects $I_1$, for all vertices $v$ in $\mathbf{P}_2$, then there are two vertices $v_1$ and $v_2$ with disjoint intervals $c(v_1, R - 1)$ and $c(v_2, R - 1)$; otherwise, $I_2 = \bigcap_{v \in \mathbf{P}_2} c(v, R - 1)$ is non-empty and intersects $I_1$, which implies that there exists a point in the intersection of $I_1$ and $I_2$ that reaches all vertices with $R - 1$ links in contradiction to the definition of the radius of $\mathbf{P}$. If $c(v_1, R - 1)$ is above $c(v_2, R - 1)$ and $p$ is a point between them, then $d(v_1, I_{11}) \geq R$ if $I_{11}$ is the part of $I_1$ below $p$; see Figure 9. Similarly, $d(v_2, I_{12}) \geq R$ if $I_{12}$ is the part of $I_1$ above $p$.

Hence, we have to solve the following subproblem. Given a subinterval $I_{ii}$ of $I_1$ such that there is a vertex $\mathbf{v}$ in $\mathbf{P}_2$ with distance $R$ to $I_{ii}$, compute $lc(\mathbf{P}) \cap a_2(I_{ii}, \mathbf{P}_2)$. Since the only property of $I_{ii}$ we will make use of is that all points on $I_{ii}$ can reach any point in $\mathbf{P}_1$ with $\varepsilon_1$ links, we assume for simplicity that $I_{ii} = I_1$. 27
We denote the maximal histogram of $I_1$ in $P_2$ by $H_1$. Note that the 2-area of $I_1$ is the union of $H_1$ with the maximal histograms of the windows of $H_1$. If $w$ is a window of $H_1$, we denote the maximal histogram of $w$ that does not intersect $H_1$ by $H_w$.

Let $\overline{w}$ be the window of $H_1$ that cuts off the part of $P_2$ that $\overline{w}$ belongs to. Without loss of generality we assume that $\overline{w}$ is a south window. Note that $\overline{w}$ needs at least $d(\overline{w}, I_1) - 1$ links to reach $\overline{w}$ since otherwise $d(\overline{w}, I_1) < R$; see Figure 9.

$\overline{w}$ splits $P_2$ into two subpolygons, the polygon that is below $\overline{w}$ which we denote by $\overline{P}_b$ and the polygon above $\overline{w}$ which we denote by $\overline{P}_a$. Note that the only part of the 2-area of $I_1$ that is contained in $\overline{P}_b$ is the maximal histogram $H_{\overline{w}}$ below $\overline{w}$. Since the part of the link center in $H_{\overline{w}}$ can be computed in linear time by Corollary 3.2, we only need to deal with $\overline{P}_a$.

**Lemma 5.4** $\overline{P}_a \cap lc(P)$ is contained in the 2-area of $\overline{w}$.

**Proof:** Let $p$ be a point in $\overline{P}_a \cap lc(P)$. Since as we noted above $d(\overline{w}, \overline{w}) \geq R - 1$ and $d(\overline{w}, p) \leq R$, Corollary 2.1 yields that $d(p, \overline{w}) \leq 2$. \qed

Let $\overline{H}$ be the maximal histogram above $\overline{w}$. An immediate consequence of Lemma 5.4 is that we can discard all the maximal histograms of south windows of $H_1$ and also those of the north window of $H_1$ which do not intersect $\overline{H}$. Furthermore, since we can compute the part of the link center in $\overline{H}$ in linear time, we only need to concern ourselves with windows $w$ that are not completely contained in $\overline{H}$. We say that window $w$ of $H_1$ is **partially**
Figure 10: The definition of $\mathbf{H}$, $e_w$, and $\mathbf{P}_1^w, \ldots, \mathbf{P}_5^w$.

visible from $\overline{w}$ if $w$ intersects $\mathbf{H}$ but is not contained in it. Since there is at most one partially visible north window $w$ whose left end point extends to the left of $\overline{w}$ and $\mathcal{L}(\mathbf{P}) \cap \mathbf{H}_w$ can be computed in linear time, we assume from now on that all partially visible north windows do not extend to the left of $\overline{w}$. We denote the family of these windows by $\mathcal{W}$.

We now divide $\mathbf{P}_2$ into five subpolygons $\mathbf{P}_1^w, \ldots, \mathbf{P}_5^w$, for each $w \in \mathcal{W}$. In order to define these subpolygons we denote the leftmost vertical edge of $\mathbf{H}_1$ that is below $w$ by $e_w$. For illustration refer to Figure 10. Since $w$ is partially visible from $\overline{w}$, $e_w$ has two reflex vertices. Let $t_w$ be the chord in $\mathbf{H}_1$ from the upper end point of $e_w$ to $w$ and $u_w$ be the chord from the lower end point of $e_w$ to $\overline{w}$. Furthermore, let $s_w$ be the horizontal line segment from the left end point of $w$ to $I_1$ and $\pi$ be the horizontal line segment from the left end point of $\overline{w}$ to $I_1$. The subpolygons are now defined as follows. $\mathbf{P}_1^w$ is defined as the subpolygon below $\pi$. $\mathbf{P}_2^w$ is defined as the subpolygon below $\overline{w}$. Note that $\mathbf{P}_1^w$ and $\mathbf{P}_2^w$ are independent of $w$. $\mathbf{P}_3^w$ is defined as the subpolygon to the right of $u_w$ and above $\overline{w}$. $\mathbf{P}_4^w$ is the subpolygon to the right of $t_w$ and the subpolygon above $w$. Finally, $\mathbf{P}_5^w$ is the subpolygon above $s_w$. Note that a point $p$ in $\mathbf{H}_w$, with $w \in \mathcal{W}$, belongs to the link center of $\mathbf{P}$ if and only if $d(p, v) \leq R$, for all $v$ in $\mathbf{P}_1^w$, $1 \leq i \leq 5$.

The first step in computing the part of the link center in $\mathbf{H}_w$ is to show that either all the points in the intersection of the 2-area of $\overline{w}$ and $\mathbf{H}_w$ can
reach the vertices of $P_i^w$ with $R$ links or none of them. A similar statement is true for the subpolygons $P_3^w$ and $P_5^w$. We will use this fact to compute a set of windows $w_1, \ldots, w_k$ such that all the points in $H_{w_i}$ can reach the vertices in $P_1^{w_i}$, $P_3^{w_i}$ and $P_5^{w_i}$ with $R$ links, $1 \leq i \leq k$.

In a second step we then compute the region inside each $H_{w_i}$ that can be reached from the vertices in $P_2^{w_i}$ and $P_4^{w_i}$ with $R$ links which yields the link center.

The subpolygon $P_1^w$

We first consider the vertices of $P_1^w$. Since we assume that the left end point of $w$ is to the right of the left end point of $\overline{w}$, the maximal histogram $H_{\overline{w}}$ above $\overline{w}$ does not intersect $w$. If there is a a vertex $v$ of $P_1^w$ such that $d(v, \overline{w}) \geq R - 1$, then, by a similar argument as in the proof of Lemma 5.4, we obtain that $H_w$ does not contain a point of $lc(P)$ and we can disregard all windows in $\mathcal{W}$. On the other hand, if $d(v, \overline{w}) \leq R - 2$, for all vertices $v$ in $P_1^w$, then any point $p$ in $H_w$ can reach a vertex $v$ in $P_1^w$ with $R$ links and we can disregard the vertices in $P_1^w$ in this case.

The subpolygon $P_3^w$

Now let $v_3$ be a vertex in $P_3^w$ and let $w_3$ be the window of $H_1$ that cuts off the subpolygon $P_{w_3}$ of $P_2$ which contains $v_3$. The next lemma shows that $H_w \setminus \overline{H}$ contains a part of the link center if and only if $v_3$ is close enough to $w_3$.

**Lemma 5.5** If $p$ is a point in the intersection of $H_w \setminus \overline{H}$ and the 2-area of $\overline{w}$, then $d(v_3, p) \leq R$ if and only if $d(v_3, w_3) \leq R - 3$.

**Proof:** We first show that if $p \in H_w \setminus \overline{H}$ and $d(v_3, p) \leq R$, then $d(v_3, w_3) \leq R - 3$. Since $p$ is not in $\overline{H}$, $p$ is to the right of $t_w$; $w_3$ is to the right of $u_w$ since $w_3 \in P_3^w$. Hence, an admissible path from $p$ to $w_3$ consists of at least four links. If $v_3$ is a vertex in the subpolygon $P_{w_3}$ of $w_3$, then Corollary 2.1 implies that $d(v_3, w_3) \leq d(p, v_3) - d(p, w_3) + 1 \leq R - 3$ since $d(p, w_3) \geq 4$.

On the other hand, if $v_3$ can reach $w_3$ with $R - 3$ links, then $d(u_w, v) \leq R - 2$ and any point in the intersection of the 2-area of $\overline{w}$ and $H_w$ has a distance of at most $R$ to $v_3$. □

Let $u_1$ be the lowest window of $H_1$ with $d(v, u_1) > R - 3$, for some vertex $v \in P_{w_1}$. By the above lemma we can discard the windows $w$ in $\mathcal{W}$ where
\(e_w\) is above \(w_i\). On the other hand, if \(w\) is a window in \(W\) such that \(e_w\) is below \(w_i\), then the above lemma implies that the vertices in \(P^w_5\) can be disregarded. Since we can find all the edges \(e_w\), where \(w\) is a window of \(H_1\), in time \(O(1)\), the windows \(w\) in \(W\) with \(e_w\) above \(w_i\) can be computed in linear time. These will not be considered further.

The subpolygon \(P^w_5\)

We now turn to the vertices of \(P^w_5\).

**Lemma 5.6** If \(p\) is a point in \(H_w\) and \(v\) a vertex of \(P^w_5\), then we have \(d(p, v) \leq R\) if and only if \(d(v, s_w) \leq R - 2\).

**Proof:** Let \(v\) be some vertex of \(P^w_5\) and \(p\) a point in \(H_w\). Since \(p\) can reach any point on \(s_w\) with two links, we have \(d(p, v) \leq R\) if \(d(v, s_w) \leq R - 2\). On the other hand, at most the second to last link of a shortest path from \(v\) to \(p\) intersects \(s_w\). Hence, \(d(v, s_w) \leq d(p, v) - 2 \leq R - 2\). \(\square\)

It is straightforward to compute the partially visible north windows of \(H_1\) that satisfy the conditions of Lemma 5.6 in linear time. Thus, we obtain a set of north windows \(w_1, \ldots, w_k\) with the following properties. The north windows \(w_1, \ldots, w_k\) are partially visible to \(\overline{w}\) and a point in the maximal histogram \(H_{w_1}\) above \(w_1\) can reach all the vertices that belong to \(P^{w_1}_1\), \(P^{w_1}_3\), and \(P^{w_1}_5\) with \(R\) links.

The link center can be viewed as the region that all vertices can reach with \(R\) links. Above we have shown that the vertices of three subpolygons can reach the points in \(H_{w_i}\) with \(R\) links, \(1 \leq i \leq k\). So in the remaining part of this section we have to show how to compute the regions of points in \(H_{w_i}\) that can be reached by the vertices in \(P^{w_i}_2\) and \(P^{w_i}_4\) with \(R\) links. Taking the intersection of the region for \(P^{w_i}_2\) and the region for \(P^{w_i}_4\) then yields the part of the link center in \(H_{w_i}\). In the following let \(H_i = H_{w_i} \setminus \overline{H}\).

**The vertices in subpolygon \(P^{w_i}_2\)**

Let \(v_l\) be the vertex in \(P^{w_i}_2\) whose interval \(I_l = \overline{w}(v_l, R-1)\) has the rightmost left end point on \(\overline{w}\) of all intervals \(\overline{w}(v, R-1)\) with \(v\) in \(P^{w_i}_2\). Similarly, let \(v_r\) be the vertex in \(P^{w_i}_2\) whose interval \(I_r = \overline{w}(v_r, R-1)\) has the leftmost right end point.

**Lemma 5.7** If \(p\) is a point in \(H_i\), then \(d(p, v) \leq R\), for all vertices \(v\) in \(P^{w_i}_2\), if and only if \(d(p, I_l) \leq 2\) and \(d(p, I_r) \leq 2\).
Figure 11: $I$ intersects $I_l = \overline{w(v_l, R - 1)}$ and $I_r = \overline{w(v_r, R - 1)}$.

Proof: Let $p$ be a point in $H_i$ and let $I$ be $\overline{wp(2)}$. We denote the left end point of $I$ by $q_1$ and the right end point by $q_2$. For illustration see Figure 11a. To show the “if” direction we assume that $I_l$ and $I_r$ intersect $I$. Then, all the right end points of the $(R - 1)$-intervals of vertices below $w$ are to the right of $q_1$ and the left end points are to the left of $q_2$. This implies that $I$ intersects all the $(R - 1)$-intervals of vertices in $P_{2}^{w_i}$. Hence, $d(p, v) \leq R$, for all vertices $v$ in $P_{2}^{w_i}$. If $p$ cannot reach one of $I_l$ or $I_r$ with two links, then the shortest path from $p$ to either $v_l$ or $v_r$ is longer than $R$ by Corollary 2.1.

We now describe how to compute the 2-area of $I_l$ and $I_r$ in $H_i$. First consider $I_l$. Clearly, the part $w^l_i$ of $w_i$ that can be reached from $I_l$ with one link can be computed in linear time using the information provided by the edge $e_{w_i}$; see Figure 11b. All the points in $H_i$ that can reach $I_l$ with two links belong to the 2-area of $w^l_i$ in $H_i$ which can be computed in time proportional to the size of $H_i$. Similarly, we compute $w^r_i$ for $I_r$ and its 2-area. Let $R_{2}^{w_i}$ be intersection of the 2-areas of $w^l_i$ and $w^r_i$ in $H_i$. Since the 2-areas of $w^l_i$ and $w^r_i$ are horizontally monotone, we can compute $R_{2}^{w_i}$ also in time $O(|H_i|)$. Hence, the total amount of time spent for computing all $R_{2}^{w_i}$, for $i = 1, \ldots, k$, is $O(\sum_{i=1}^{k} |H_i|) = O(n)$.

The vertices in subpolygon $P_{4}^{w_i}$

Finally, consider the vertices in $P_{4}^{w_i}$. Since $P_{4}^{w_i}$ contains $H_i$ we can apply Theorem 1 to $P_{4}^{w_i}$ and compute the part $R_{1}^{w_i}$ of $H_i$ that can be reached.
by all vertices in \( P^{w_i}_4 \) with \( R \) links in time proportional to \( |P^{w_i}_4| + |H_{w_i}| \). Note that \( R \) is different from the radius of \( P^{w_i}_4 \) but Theorem 1 can be applied nevertheless. Again the resulting region \( R_i^{w_i} \) is monotone and of size \( O(\|P_4^{w_i}\|) \). Since the subpolygons \( P^{w_i}_4 \) are disjoint from each other, the amount of time spent for this step is

\[
O\left(\sum_{i=1}^{k} |P^{w_i}_4| + |H_{w_i}|\right) = O(\|P_2\|) = O(n).
\]

So we are left with the task of intersecting the regions \( R^{w_i}_2 \) and \( P^{w_i}_4 \) both of which are monotone with respect to the \( x \)-axis and of size \( O(\|P^{w_i}_4\|) \), for \( i = 1, \ldots, k \). This yields the part of the link center of \( P \) that is contained \( H_{w_i} \). Since intersecting two monotone polygons requires only time linear in the size of the polygons, this last step again takes time \( O(\sum_{i=1}^{k} |P^{w_i}_4|) = O(n) \) which shows that the part of the link center in the 2-area of \( I_1 \) can be computed in linear time. This completes the algorithm to compute the part of the link center of \( P \) in the 2-area of \( I_1 \) in \( P_2 \). It is summarized below.

We have shown the following theorem.

**Theorem 4** If \( P \) is a simple rectilinear polygon, then the rectilinear link center of \( P \) can be computed in time linear in the number of vertices of \( P \).

### 6 Conclusions

In this paper we have studied the concept of rectilinear link distance within simple rectilinear polygons. We have given an algorithm to compute the rectilinear link center. This algorithm runs in linear time and employs covering technique for the relevant part of the polygon consisting of histograms such that the link center can be computed efficiently. To our knowledge this is the first optimal algorithm for a center problem.

An open problem is, of course, to try to generalize the technique for polygons and path links that come from some fixed set of orientations. It may be possible to improve on the \( O(n \log n) \) time upper bound which exists for general polygons. Here, one problem that arises is that the regions that have to be intersected may no longer be monotone with respect to a common line.

In a rectilinear world consisting of axis parallel line segments it is also possible to define the link center. There exists a naive \( O(n^4) \) time algorithm to solve the link center problem in this case and a lower bound of \( \Omega(n^2) \) time
Algorithm \textit{Link-Center-in-} \textit{a}_2(\textit{I}_1, \textit{P}_2) \\
\textbf{Input:} A simple rectilinear polygon \textit{P}, a central chord \textit{c} with vertices at distance at least \(R-1\) to both sides, and an interval \textit{I}_1 on \textit{c}. \\
\textbf{Output:} The link center of \textit{P} in \textit{a}_2(\textit{I}_1, \textit{P}_2) where \textit{P}_2 is on one side of \textit{c}.

1. Split \textit{I}_1 into two subintervals \(\textit{I}_{11}\) and \(\textit{I}_{12}\) such that \(c(\textit{I}_{1i}, \textit{P}_2) \geq R\), for \(i = 1, 2\).
2. Compute \(\overline{w}, \overline{e}, \textit{H}_{\overline{w}}, \textit{H}\).
3. Apply the subroutine \textit{Part-in-Histogram} to compute the part of \(\textit{lc(P)}\) in \(\textit{H}_{\overline{w}}\), \(\textit{H}\).
4. \textbf{if} there is a vertex \(v \in \textit{P}_1^w\) with \(d(v, \overline{e}) > R - 1\) \textbf{then} \textbf{output} \#.
5. \textbf{else} \(\textit{W} := \text{set of windows partially visible from } \overline{w}\).
6. \(w_l := \text{the lowest window } w \text{ with } d(v, w) > R - 3, \text{ for a vertex } v \in \textit{P}_w\).
7. \(\textit{W}_l := \text{set of north windows above } w_l\).
8. Compute windows \(w_1, \ldots, w_k \in \textit{W} \setminus \textit{W}_l\) such that \(d(v, s_{w_i}) \leq R - 2\), for all \(v \in \textit{P}_5^{w_i}\).
9. \(I_l := \overline{w}(w_l, R - 1); I_r := \overline{w}(v_r, R - 1)\).
10. \textbf{for} all windows \(w_l\) \textbf{do}
    11. \(R_2^{w_l} := \textit{a}_2(\textit{I}_l, \textit{P}) \cap \textit{a}_2(\textit{I}_r, \textit{P}) \cap (\textit{H}_{w_i} \setminus \textit{H}_{\overline{w}})\).
    12. Apply the subroutine \textit{Part-in-Histogram} to compute the region \(R_1^{w_l}\) for \(\textit{H}_{w_i} \setminus \textit{H}_{\overline{w}}\) in the polygon given by \(\textit{P}_4^{w_l}\) with parameter \(R\).
13. \textbf{endfor}
14. \textbf{output} \(\bigcup_{1 \leq i \leq k}(R_2^{w_l} \cap R_4^{w_l})\).
15. \textbf{endif}

End \textit{Link-Center-in-} \textit{a}_2(\textit{I}_1, \textit{P}_2)

since the link center may consist of a quadratic number of regions. It would be of interest to tighten these two bounds, i.e., to present faster algorithms and, if possible, to exhibit better lower bounds.

Acknowledgements

We thank the anonymous referees for the careful reading of the paper and for their support in making it more readable which has not only improved the presentation of the results but also our understanding of the problem. We in particular thank them for pointing out an error in Section 3.3.1. We would also like to thank Th. Ottmann for the many discussions we had on
preliminary versions of the result.

References


