Partitioning Polygons into Tree Monotone and Y-Monotone Subpolygons

Ralph P. Boland† and Jorge Urrutia‡

†University of Waterloo rpboland@uwaterloo.ca
‡Instituto de Mathemáticas, Universidad Nacional Autónoma de México urrutia@matem.unam.mx

Abstract A polygon Q is tree monotone if, for some highest or lowest point p on Q and for any point q interior to Q, there is a y-monotone curve from p to q whose interior is interior to Q. We show how to partition an n vertex polygon P in $\Theta(n)$ time into tree monotone subpolygons such that any y-monotone curve interior to P intersects at most two of the subpolygons. We then use this partition to further partition P into y-monotone subpolygons such that the number of subpolygons needed to cover any given y-monotone curve interior to P is $O(\log n)$. Our algorithm runs in $\Theta(n)$ time and space which is an improvement by an $O(\log n)$ factor in time and space over the best previous result.

Keywords: tree monotone partition, monotone cover number, circular ray shooting

1 Introduction

The monotone cover number of a decomposition Π of an n vertex simple polygon P into (possibly overlapping) subpolygons, denoted $\mathtt{C}_{\wr}(\Pi)$, is the smallest number k such that any y-monotone curve contained in P can be covered by at most k of the subpolygons and their interiors [2]. A Hierarchical Vertical Decomposition [6] of P (modified to use a horizontal visibility map instead of a vertical visibility map [8]) contains a set D of overlapping y-monotone subpolygons of P such that $\mathtt{C}_{\wr}(D) = O(\log n)$ and D requires $O(n \log n)$ storage.

We present two algorithms that also construct decompositions of P into y-monotone subpolygons such that the monotone cover number is $O(\log n)$. These algorithms have the advantages that they produce decompositions requiring $\Theta(n)$ storage, they run in $\Theta(n)$ time, and the decompositions of P constructed are partitions; that is the interiors of the subpolygons do not overlap.

We first present a partition of P of interest in its own right. We call a y-monotone curve whose interior is interior to P a *y-curve* of P. We call P *tree monotone* if there is a highest or lowest point p on P such that, for any point q interior to P, there is a y-curve of P from p to q. We call a partition, say Π, of P by chords a *tree monotone partition* of P if the following two conditions hold (opposition is defined in Section 3):

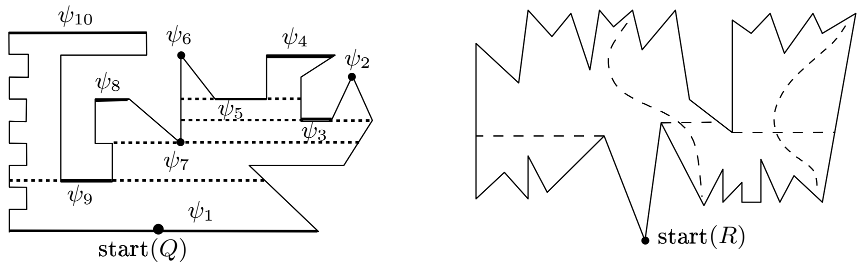

Figure1. *Left: A tree $+y$-monotone polygon Q, its cusps ($\psi_1 - \psi_{10}$), and its lids. Right: A tree monotone partition of a polygon R. Note that the interior of any y-curve of R intersects at most two subpolygons of the partition*

1. The subpolygons of Π are tree monotone.
2. Any pair of coincident subpolygons of Π are in opposition.

The key property of any tree monotone partition Π of P (shown to hold in Section 3) is that $\mathsf{C}_\wr(\Pi) \leq 2$. This property can sometimes be used to reduce a problem on polygons to the special case of polygons partition-able into two tree *monotone* subpolygons in opposition. This is the approach taken in [2, 3] to develop a solution to the circular ray shooting problem [1].[1] We present a simple algorithm (given the horizontal visibility map of P) that constructs a tree monotone partition of P in $\Theta(n)$ time.

We then present two algorithms that use this tree monotone partition of P to further partition P into y-monotone subpolygons such that the monotone cover number of the partition is $O(\log n)$. The first algorithm constructs a partition of P into y-monotone subpolygons by chords whose monotone cover number is within two of the minimum monotone cover number of any partition of P into y-monotone subpolygons by chords. The second algorithm pays more attention to the sizes of the y-monotone subpolygons of the partition. It produces a partition such that any given y-curve of P can be broken into two halves such that each half is covered by a sequence of $O(\log n)$ subpolygons (and their interiors) of the partition and the number of vertices of the subpolygons in each sequence decreases exponentially. This property of the partition is used in [2,3] to develop a solution to the circular ray shooting problem with the same space and query complexity as the best previous result [6] but which is simpler.[2]

The remainder of this paper is organized as follows. In Section 2 we develop some basic tools that we need. In Section 3 we establish a properties of tree monotone polygons and tree monotone polygon partitions and use them to develop an algorithm that builds a tree monotone partition of P. In Section 4 we use a tree monotone subpolygon partition of P to construct the y-monotone subpolygon partitions of P described in the previous paragraph.

[1] The circular ray shooting problem is to preprocess P so that, given a query directed circular arc ζ, one can quickly find the first intersection of ζ with P.

[2] The preprocessing complexity was also improved by an $O(\log n)$ factor.

2 Some Notation And Tools

Let P be an n vertex simple polygon with vertices $\{v_0, v_1, \ldots, v_{n-1}\}$ in counter-clockwise order and edges $\{e_o, \ldots, e_{n-1}\}$ such that v_i and v_{i+1} are the vertices of e_i addition taken modulo n. We will use v_j to denote $v_{(j \bmod n)}$. To simplify the discussion we assume no adjacent edges of P are collinear. We denote the interior of P by interior(P) and denote region(P) = interior(P) $\cup$ P. By partition of P we mean a set of polygons whose interiors do not intersect and whose regions together contain region(P).

A cusp ψ of P is either a vertex v_i of P such that v_{i-1}, and v_{i+1} lie on the same side of the horizontal line through v_i, or a horizontal edge e_i of P such that v_{i-1} and v_{i+2} lie on the same side of the line through e_i. We say that ψ is a ***reflex (convex) cusp*** if it is either a reflex (convex) vertex or an edge whose endpoints are both reflex (convex) vertices. For any point or segment e, mid(e) denotes e if e is a point and the midpoint of e if e is a segment.

In this paper any subpolygon R of P, including P itself, has a special point we call its ***start point*** and which we denote by start(R). We assume start(P) = mid(ψ) where ψ is some convex cusp of P. The choice of start(P) has no impact on the results of this paper but it does cause the algorithms presented here to produce slightly different partitions. The chords of P used in this paper are assumed to not intersect start(P). For any chord γ of P we denote by P_γ the subpolygon of P, of the two determined by γ, that does not contain start(P).

Let Q be any simple subpolygon of P such that each edge of Q is the union of one or more chords, edges, or parts of edges of P. Then we call Q a ***cut subpolygon*** of P. All subpolygons in this paper will be cut subpolygons. When we say that a path is interior to P we mean its interior is contained in interior(P). By the ***door*** of Q, denoted door(Q), we mean the chord of P on an edge of Q that must be crossed by every path interior to P between any interior point of Q and start(P); if start(P) is a point on Q then the cusp of P containing start(P) is door(Q). By the ***windows*** of Q, denoted windows(Q), we mean the remaining chords of P that are edges of Q or parts thereof. We assume that start(Q) = mid(door(Q)).

Let Π be a partition of P into cut subpolygons. Then we call Π a ***tree partition*** of P. This term is appropriate because the graph in which each vertex corresponds to an subpolygon of Π and the edges of the graph correspond to coincident pairs of subpolygons of Π is a tree. All partitions used in this paper are tree partitions. Let R and S be two subpolygons of Π such that door(S) $\in$ windows(R). Then we say that R is the ***parent subpolygon*** of S and S is a ***child subpolygon*** of R. For any subpolygon U of Π we denote by parent(U) the parent of U if U has a parent and the value nil otherwise.

We denote by $\Pi_\boxminus(P)$ the tree partition of P into trapezoids by the horizontal visibility map of P. To simplify the discussion we assume $\Pi_\boxminus(P)$ contains at least two trapezoids. We may abbreviate the term horizontal chord to *h-chord*. Let t be any trapezoid of $\Pi_\boxminus(P)$ and p be the midpoint of the h-chord whose endpoints are the midpoints of the non-horizontal edges of t. If some convex cusp ψ of P is a top or bottom edge of t then let q = mid(ψ); otherwise let $q = p$. We denote mid(t) = q. Let $\mathcal{T}$ be the dual tree of $\Pi_\boxminus(P)$ such that the root node of $\mathcal{T}$ corresponds to the subpolygon of $\Pi_{\boxminus}(P)$ that contains start(P) and each edge of $\mathcal{T}$ is directed towards its parent node. We denote by $\Lambda(P)$ the plane embedding of $\mathcal{T}$ such that:

1. For each trapezoid $t \in \Pi_{\boxminus}(P)$, mid(t) is the vertex of $\Lambda(P)$ corresponding to the the dual vertex of t in $\mathcal{T}$.
2. For any two trapezoids t and t_p of $\Pi_{\boxminus}(P)$ such that parent$(t) = t_p$, the chain $C = \mathrm{mid}(t), \mathrm{mid}(\mathrm{door}(t)), \mathrm{mid}(t_p)$ corresponds to the edge of $\mathcal{T}$ joining the vertices of $\mathcal{T}$ corresponding to t and t_p. Chain C is directed upwards if mid(t_p) is above mid(t) and downwards otherwise.
3. Each vertex of $\Lambda(P)$ is colored pink (brown) if its outgoing edge is directed upwards (downwards); the root vertex of $\Lambda(P)$ is given the color shared by its child vertices.

Note that $\Lambda(P)$ intersects each h-chord of P and start(P) is its root vertex; see Fig. 2.

3 Tree Monotone Polygons and Partitions

Many properties of tree monotone polygons are established in [2]. Here we need the following result.

Lemma 1. *P is tree monotone if and only if all vertices of $\Lambda(P)$ have the same color.*

Proof. $\Rightarrow$ Consider otherwise that there are two differently colored and adjacent vertices u_p and u_c of $\Lambda(P)$ such that u_p is the parent of u_c. Let t_p and t_c be the trapezoids of $\Pi_{\boxminus}(P)$ containing u_p and u_c respectively. Then some horizontal line ℓ contains both door(t_p) and door(t_c) since u_p and u_c have different colors. Let C be a y-curve of P from start(P) to u_c; C exists since P is tree monotone. Now C intersects both door(t_p) and door(t_c) so C intersects ℓ twice which is impossible.

$\Leftarrow$ We will show that any point p interior to P and start(P) are the endpoints of some y-curve of P. Let q be a point on $\Lambda(P)$ and $\wp$ be the path on $\Lambda(P)$ from start(P) to q such that some h-chord of P contains $\overline{pq}$ and $\overline{pq} \cap \wp = q$. Then the path $\wp \cup \overline{qp}$ forms a y-curve of P since all the vertices of $\Lambda(P)$ have the same color and thus the edges joining them have the same orientation.

Let ψ be a reflex cusp of P. Then we call the two h-chords of P with an endpoint on ψ the ***lids*** of ψ. We denote $\mathfrak{L}(P) = \{\gamma : \gamma \text{ is a lid of some reflex cusp of } P\}$. We use $\pi(P)$ to denote the partition of P induced by $\mathfrak{L}(P)$. Clearly, the subpolygons of $\pi(P)$ are y-monotone. Let ξ be the subset of edges of $\Lambda(P)$ whose endpoints are colored differently. We denote by $\mathfrak{L}_r(P)$ the subset of elements of $\mathfrak{L}(P)$ intersected by some element of ξ; note that $|\mathfrak{L}_r(P)| = |\xi|$. We denote by $\pi_r(P)$ be the partition of P induced by $\mathfrak{L}_r(P)$. We now show that $\pi_r(P)$ is a tree monotone partition of P.

Observation 1. *Let Q be a subpolygon of P such that $\Pi_{\boxminus}(Q) \subset \Pi_{\boxminus}(P)$. Then, for any trapezoid t in $\Pi_{\boxminus}(Q)$, vertex mid(t) has the same color in $\Lambda(Q)$ as in $\Lambda(P)$ (but not necessarily the same location).*

If P is tree monotone and start(P) is a lowest (highest) point of P then we call P *tree $+y$-monotone* (*tree $-y$-monotone*). We say that two tree monotone subpolygons of P are *in opposition* if one is $+y$-monotone and the other is $-y$-monotone and the door of one is a window of the other. The following theorem establishes the key property of tree monotone partitions.

Theorem 1. *Let Π be any tree monotone partition of P. Then the interior of any y-curve of P intersects at most two subpolygons of Π.*

Proof. Assume otherwise that the interior of some directed y-curve C of P intersects three different subpolygons $Q,R,T \in \Pi$ in succession. We consider only the case that door(Q) and door(T) are both windows of R. The remaining case, that door(R) is a window of either Q or T, is handled similarly; see [2]. Let $u_Q = \Lambda(P) \cap \text{door}(Q)$ and $u_T = \Lambda(P) \cap \text{door}(T)$. Let $\wp_Q$ be the path on $\Lambda(P)$ from u_Q to start(P) and $\wp_T$ be the path on $\Lambda(P)$ from u_T to start(P). Let v be the point on $\Lambda(P)$ that is the endpoint of $\wp_Q \cap \wp_T$ other than start(P). Then $v =$ mid(t) where t is some trapezoid of $\Pi_{\boxminus}(P)$. Let $\wp_Q^v$ be the portion of $\wp_Q$ from u_Q to v and $\wp_T^v$ be the portion of $\wp_T$ from u_T to v. By Lemma 1, $\wp_Q^v$ is y-monotone and the edges of $\wp_Q^v$ have the same orientation, either upwards or downwards. Similarly the edges of $\wp_T^v$ have the same orientation. Now Q and T are not in opposition since they are both in opposition with R so the edges of $\wp_Q^v$ and $\wp_T^v$ all have the same orientation. Thus $\wp_Q$ and $\wp_T$ enter t through different windows but from the same top or bottom edge, say e, of t. But now C must also cross both of these windows so C must cross e twice which contradicts that C is y-monotone.

We call a tree monotone cut subpolygon Q of P *maximal* if no tree monotone cut subpolygon of P with the same door as Q properly contains Q.

Lemma 2. *The subpolygons of $\pi_r(P)$ are maximal tree monotone.*

Proof. Consider any subpolygon Q of $\pi_r(P)$. By construction and Observation 1, the vertices of $\Lambda(Q)$ have the same color so Q is tree monotone by Lemma 1. Now assume that there exists a subpolygon Q' of P such that region$(Q) \subset$ region(Q') and door$(Q') =$ door(Q). We will show that Q' is not tree monotone. Let t be a trapezoid of $\Pi_{\boxminus}(Q') - \Pi_{\boxminus}(Q)$ such that door$(t) \in$ windows(Q). Then region$(t) \subseteq$ region(t_P) for some trapezoid t_P of $\Pi_{\boxminus}(P)$. Now mid(t) and mid(t_P) have the same color and, by the construction of $\Lambda(P)$, this color differs from the shared color of the vertices of $\Lambda(Q)$. Therefore Q' is not tree monotone by Lemma 1.

We say that two subpolygons of a tree partition are coincident if one subpolygon is the parent of the other.

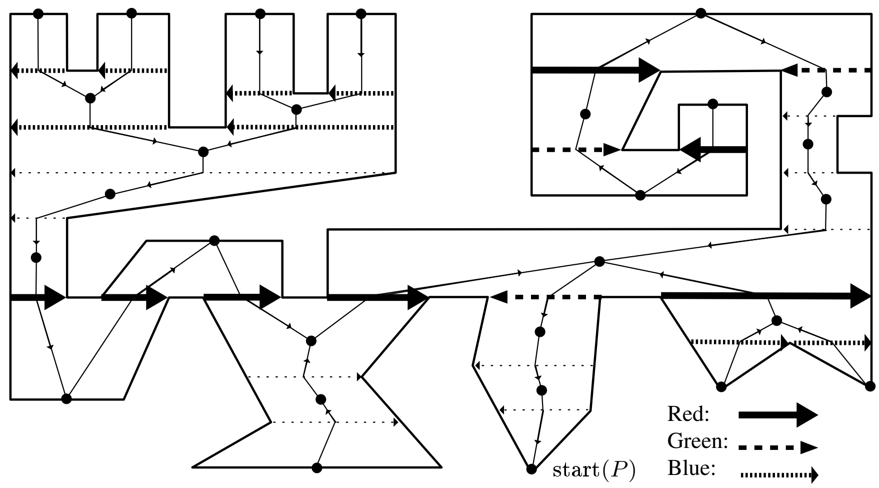

Figure2. *Tree $\Lambda(P)$ and the construction of $\mathfrak{L}_r(P)$ and $\pi_r(P)$ using Rule 1*

Lemma 3. *All coincident pairs of subpolygons of $\pi_r(P)$ are in opposition.*

Proof. Assume otherwise that $\pi_r(P)$ has a coincident pair of subpolygons, say Q_1 and Q_2, that are not in opposition. Assume, without loss of generality, that $Q_1 = \text{parent}(Q_2)$ and that both are tree $+y$-monotone. Let Q_3 be the polygon whose interior is the union of the interiors of Q_1, Q_2, and door(Q_2); note that door$(Q_3) =$ door(Q_1). Now Q_3 is tree $+y$-monotone and properly contains Q_1 which is impossible by Lemma 2.

Theorem 2. *$\pi_r(P)$ is a tree monotone partition of P.*

Proof. This follows directly from Lemmas 2 and 3.

The following result is also of interest.

Theorem 3. *Any convex set S contained in region(P) intersects the interior of at most two subpolygons of $\pi_r(P)$.*

Proof. Let T be the subset of the subpolygons of $\pi_r(P)$ that intersect S. Then S contains a directed strictly y-monotone curve C with endpoints p_1 and p_2 such that p_1 and p_2 have the maximum y-coordinate and minimum y-coordinate respectively of any point on the boundary of S, C intersects every polygon in T, and C is a y-curve of P. Now, applying Theorem 1 to C, implies that $|T| \leq 2$.

The following additional results on $\pi_r(P)$ are established in [2].

Lemma 4. *Any curve in interior(P) and decomposable into k interior disjoint y-curves of P intersects the interior of at most $k+1$ of the subpolygons of $\pi_r(P)$.*

Lemma 5. *Any curve in interior(P) and on the boundary of a convex set intersects the interior of at most four subpolygons of $\pi_r(P)$.*

For any subpolygon Q of $\pi_r(P)$ with at least one window let family(Q) be the subpolygon of P whose interior is the union of the interiors of Q, the windows of Q, and the child subpolygons of Q. Let $D_{\text{fm}}(P) = \{P\}$ if $|\pi_r(P)| = 1$ and $D_{\text{fm}}(P) = \cup_{Q_i \in \pi_r(P) \text{ and windows}(Q_i) \neq \emptyset}(\text{family}(Q_i))$ otherwise.

Theorem 4. *Any convex set in region(P) is contained in one of the elements of $D_{\text{fm}}(P)$ and the total number of vertices in all the elements of $D_{\text{fm}}(P)$ is $\Theta(n)$.*

Theorem 4 suggests an approach to the well studied problem of finding convex objects in P: search each of the elements of $D_{\text{fm}}(P)$. This will prove useful if either the elements of $D_{\text{fm}}(P)$ are substantially smaller than P or if the properties of the elements of $D_{\text{fm}}(P)$ can be exploited in some way.

We now show that $\pi_r(P)$ can be constructed in $\Theta(n)$ time. Though $\mathfrak{L}_r(P)$ (and thus $\pi_r(P)$) can be found by first constructing $\Lambda(P)$, it is more easily found directly. For any window, say w, of any subpolygon Q of any tree partition Π of P we denote door$(\Pi, w) =$ door(Q). We now assign each of the elements of $\mathfrak{L}(P)$ one of the colors red, blue, and green such that the red ones constitute $\mathfrak{L}_r(P)$; see Fig. 2. Each element of $\mathfrak{L}(P)$ is initially colored blue. Next the following rule is applied to each chord $\gamma \in \mathfrak{L}(P)$ in any order:

Rule 1. Let $\sigma =$ door$(\Pi_{\boxminus}(P), \gamma)$. If γ and σ are collinear then 1) color γ red and 2) color σ green if it is colored blue but not if it is colored red.

Theorem 5. *$\mathfrak{L}_r(P)$ is the subset of the elements of $\mathfrak{L}(P)$ that are red.*

Proof. Let u,v, and w be vertices of $\Lambda(P)$ and d and e be edges of $\Lambda(P)$ such that u, d, v, e, w is a path of $\Lambda(P)$ directed towards the root of $\Lambda(P)$. Let t_u, t_v, and t_w be the trapezoids of $\Pi_{\boxminus}(P)$ containing u,v, and w respectively. Then d crosses door(t_u) and e crosses door(t_v). By Rule 1, door(t_u) is colored red if and only if door(t_u) and door(t_v) are collinear. Also, door$(t_u) \in \mathfrak{L}_r(P)$ if and only if one of d and e are directed upwards and the other is directed downwards. But this occurs if and only if door(t_u) and door(t_v) are collinear so door$(t_u) \in \mathfrak{L}_r(P)$ if and only if it is red. Now u, d, v, e, w was an arbitrary path of $\Lambda(P)$ so door(t_u) can be any element of $\mathfrak{L}(P)$ except the elements of S where S is the subset of $\mathfrak{L}(P)$ that cross an edge of $\Lambda(P)$ connected to start(P). But any element λ of S is neither red nor collinear with door$(\pi(P), \lambda) =$ door(P) so $\mathfrak{L}_r(P)$ is exactly the set of red elements of $\mathfrak{L}(P)$.

Corollary 1. *[2] The blue lids of $\mathfrak{L}(P)$ are the lids of the subpolygons of $\pi_r(P)$ and the green lids of $\mathfrak{L}(P)$ are chords but not lids of the subpolygons of $\pi_r(P)$.*

Now let ψ be any reflex cusp of P. If the lids of ψ are not lids of any other cusp of P then an even simpler rule can be used to color them. Also, if the dual tree of $\Pi_{\boxminus}(P)$ is not available then Rule 1 can be modified so that the door of each trapezoid can be found using an orientation of the chords of the horizontal visibility map of P. This orientation is shown in Fig. 2. For details of these modifications to Rule 1 see [2].

Theorem 6. *$\mathfrak{L}_r(P)$ and $\pi_r(P)$ can be constructed in $\Theta(n)$ time.*

Proof. In $\Theta(n)$ time $\Pi_{\boxminus}(P)$ can be constructed [4].[3] Then, the red chords of $\mathfrak{L}(P)$ can be found in $\Theta(n)$ time using $\Pi_{\boxminus}(P)$ and Rule 1. But these red chords are just $\mathfrak{L}_r(P)$ by Theorem 5. Finally, $\pi_r(P)$ is easily constructed from $\mathfrak{L}_r(P)$ and $\Pi_{\boxminus}(P)$ in $\Theta(n)$ time.

A variation of our algorithm for constructing $\pi_r(P)$ deals with polygons with holes. It constructs a partition of a polygon with holes into tree monotone subpolygons with holes such that the pockets of the holes are also tree monotone polygons; see [2].

4 *Y*-Monotone Partitions

We now present an algorithm for constructing a partition of a tree monotone polygon into y-monotone subpolygons such that the monotone cover number of the partition is minimized. Using this algorithm we can partition a simple polygon into y-monotone subpolygons such that the monotone cover number of the partition is within two of the minimum monotone cover number of any partition of the polygon into y-monotone subpolygons. The proofs of the lemmas in this section are found in [2].

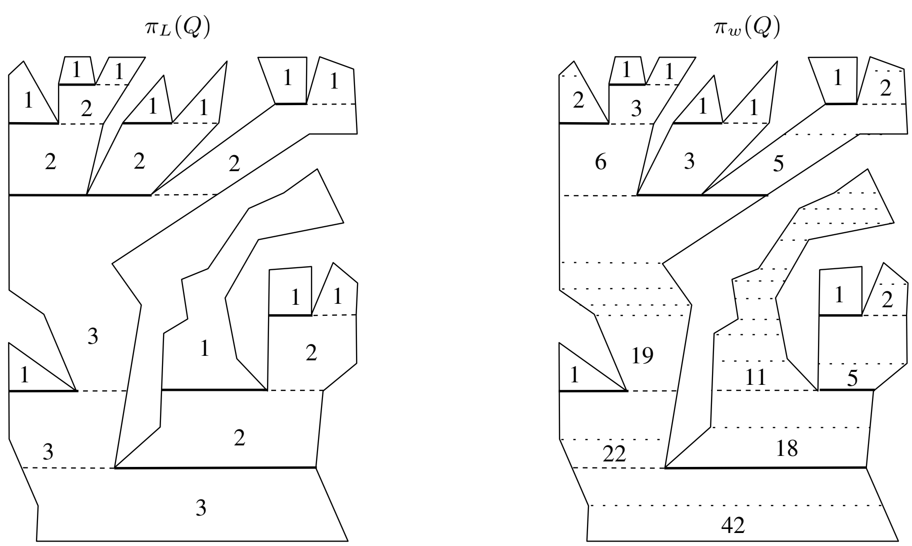

Figure3. *Partitions $\pi_L(Q)$ and $\pi_w(Q)$ of an tree $+y$-monotone polygon Q into y-monotone subpolygons; $\mathtt{C}_{\wr}(\pi_L(Q)) = 3$ and $\mathtt{C}_{\wr}(\pi_w(Q)) = 4$*

[3] There are many simpler $O(n \log n)$ time algorithms to choose from; e.g. [5, 8].

A tree, say T, can be partitioned into paths by deleting all edges on each internal node except the edge to the parent node and the edge of one child node. If, for each internal node, the edge kept is to the child node that is the root of the largest subtree the centroid path decomposition of T [7] is constructed. If, instead, for each internal node, the edge kept is to the child that is the root of the subtree of maximum height then the partition constructed is what we call the *longest path decomposition of* T. In both cases we break ties arbitrarily.

Let Q be an m vertex tree monotone polygon, T_w be the centroid path decomposition of $\Lambda(Q)$, and T_L be its longest path decomposition. We denote by $\pi_w(Q)$ the partition of Q induced by the edges of $\pi(Q)$ crossed by some edge of T_w and by $\pi_L(Q)$ the partition of Q induced by the edges of $\pi(Q)$ crossed by some edge of T_L.

Lemma 6. $\mathsf{C}_{\wr}(\pi_L(Q)) \leq \log m$ *and this value is the minimum monotone cover number of any partition of* Q *by chords.*

We denote $\pi_L(P) = \cup_{Q_i \in \pi_r(P)} \pi_L(Q_i)$ and $\pi_w(P) = \cup_{Q_i \in \pi_r(P)} \pi_w(Q_i)$; note that $\pi_L(P)$ and $\pi_w(P)$ are partitions of P.

Lemma 7. *Partitions* $\pi_L(P)$ *and* $\pi_w(P)$ *can be constructed in* $\Theta(n)$ *time.*

From Theorem 1 and Lemma 6 it is clear that $\mathsf{C}_{\wr}(\pi_L(P))$ is within a factor of two of the minimum monotone cover number of any partition of P by chords. However, this bound is actually much tighter.

Lemma 8. $\mathsf{C}_{\wr}(\pi_L(P))$ *is at most* $2\log n - 1$ *and within two (one) of the minimum monotone cover number of any partition of* P *by chords (*h*-chords).*

While this result is interesting mathematically it ignores the size of the subpolygons. For designing algorithms, partition $\pi_w(P)$ may be a better choice because of the following result. Let Π_U be a tree partition of a polygon U. We say that Π_U has the *telescoping property* if, for every subpolygon R of Π_U such that parent$(R) \neq$ nil, that $|\Pi_{\boxminus}(\text{parent}(R))| > 2\,|\Pi_{\boxminus}(R)|$.

Lemma 9. $\pi_w(Q)$ *has the telescoping property.*

Note that this lemma implies that $\mathsf{C}_{\wr}(\pi_w(Q)) \leq \log m$. More importantly, this property may allow Q to be searched more efficiently in situations where we can assign weights to the trapezoids of a y-monotone polygon so that the trapezoids with large weight are found more quickly by the search algorithm, as for example occurs with weight-balanced trees.

In [2, 3] this property of $\pi_w(Q)$ is used find a solution to the circular ray shooting problem on Q matching the space and query complexity of the best previous result [6] but with a simpler algorithm. We point out that if $\pi_L(Q)$ were used instead of $\pi_w(Q)$, the solution would have a query time of $O(\log^2 m)$. However, if $\Lambda(Q)$ has a height of $O(\log m)$ the cost of a query is $O(\log m)$ even using $\pi_L(Q)$. This matters because in this case the solution can be made much simpler and also faster by a constant factor if $\pi_L(Q)$ is used.

Finally, using Theorem 1, this solution is easily generalized to apply to simple polygons.

Perhaps this approach can also be used to develop a simpler ray shooting algorithm for polygons. This would require that a sufficiently simple algorithm for ray shooting in y-monotone polygons be found. We are exploring further applications of the results of this paper.

References

1. Pankaj K. Agarwal and Micha Sharir. Circle shooting in a simple polygon. *J. Algorithms*, 14:69–87, 1993.
2. R. P. Boland. *Polygon Visibility Decompositions with Applications.* Ph.D. thesis, March, 2002, University of Ottawa., Ottawa, Ontario, Canada, 2000.
3. R. P. Boland and J. Urrutia. A simpler circular ray shooting algorithm. In *Proc. 13th Canad. Conf. Comput. Geom.*, pages 37–40 (Electronic version 7 pages), 2001.
4. Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.
5. Bernard Chazelle and Janet Incerpi. Triangulation and shape-complexity. *ACM Trans. Graph.*, 3(2):135–152, 1984.
6. Siu-Wing Cheng, Otfried Cheong, Hazel Everett, and René van Oostrum. Hierarchical vertical decompositions, ray shooting, and circular arc queries in simple polygons. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 227–236, June 1999.
7. Richard Cole, Martin Farach-Colton, Ramesh Hariharan, Teresa Przytycka, and Mikkel Thorup. An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM J. Comput.*, 30(5):1385–1404 (electronic), 2000.
8. A. Fournier and D. Y. Montuno. Triangulating simple polygons and equivalent problems. *ACM Trans. Graph.*, 3(2):153–174, 1984.