# Finding the Medial Axis of a Simple Polygon in Linear Time

Francis Chin* Jack Snoeyink† Cao An Wang‡

**Abstract**

We give a linear-time algorithm for computing the medial axis of a simple polygon $P$, This answers a long-standing open question—previously, the best deterministic algorithm ran in $O(n \log n)$ time. We decompose $P$ into pseudo-normal histograms, then influence histograms and $xy$ monotone histograms. We can compute the medial axes for $xy$ monotone histograms and merge to obtain the medial axis for $P$.

## 1 Introduction

The *medial axis* of a simple plane polygon $P$ goes by many names, including *symmetric axis* or *skeleton*. One of the more picturesque is the *grassfire transform*: Imagine igniting all boundary points of $P$. If the flame burns inward at a uniform rate, then the *quench points* where the flame meets and extinguishes itself define the medial axis. Equivalently, the medial axis is the locus of all centers of circles inside $P$ that touch the boundary of $P$ in two or more points.

The medial axis was proposed and named by Blum [3] in 1967 article entitled, "A transformation for extracting new descriptors of shape." The pattern recognition literature uses it heavily as a one-dimensional structure that represents two-dimensional shape [7, 14, 16]; it has also been used in solid modelling [18], mesh generation [8], pocket machining [9], etc.

To a computational geometer, the medial axis of an $n$-gon $P$ is a Voronoi diagram [2, 15] whose sites are the open edges and the vertices of the boundary. In 1982, Lee [13] developed an $O(n \log n)$ algorithm to compute the medial axis.Since that time, it has been an open question to determine the time required to compute the medial axis. Up to this year there were two significant milestones: In 1987 Aggarwal et al. [1] first published an algorithm that can compute the medial axis of a *convex* polygon $P$ in linear time. In 1991 Devillers [5] first published a *randomized* algorithm for the medial axis that runs in $O(n \log^* n)$ expected time.

*Dept of Comp. Science, University of Hong Kong, Hong Kong. `chin@csd.hku.hk`

†Dept of Comp. Science, UBC, Vancouver, BC, Canada V6T 1Z4. `snoeyink@cs.ubc.ca`
Partially supported by an NSERC grant and a BC ASI Fellowship.

‡Dept of Comp. Science, Memorial University of Newfoundland, St. John's, NFLD, Canada A1C 5S7 `wang@garfield.cs.mun.ca` Work partially supported by NSERC grant OPG0041629.

Decompositions of $P$ into histograms have been applied to the *constrained Voronoi diagram* of vertices of $P$. Djijev and Lingas' [6] showed that the algorithm of Aggarwal et al. [1] could find the constrained Voronoi for an $xy$ monotone histogram. Klein and Lingas [11] gave a randomized algorithm for the constrained Voronoi of a histogram and, by merging, computed the constrained Voronoi of $P$ in $O(n)$ expected time. Wang and Chin [19] gave a deterministic algorithm by further decomposing histograms.

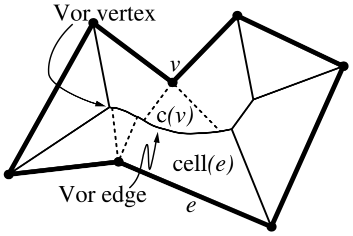

Fig. 1: Medial axis (solid) and Voronoi diagram (solid & dotted)

We extend Wang and Chin's decomposition to compute the medial axis of a simple polygon $P$. Our algorithm decomposes $P$ into normal histograms, then into influence histograms, and $xy$ monotone histograms. It computes the Voronoi diagrams of $xy$ monotone histograms, and merges to obtain the medial axis of $P$. After reviewing definitions and known results about Voronoi diagrams and histogram decompositions in Section 2, we describe the new steps in reverse order: In Section 3, we extend the algorithm of Aggarwal et al.[1] to compute the Voronoi diagram of selected edges and vertices of an $xy$-monotone histogram. In the process, we simplify part of the analysis. In Section 4, we compute the Voronoi diagram of a histogram by decomposing it into influence histograms and $xy$ monotone histograms. We conclude in Section 5.

Klein and Lingas have recently extended their work on histogram decompositions to obtain the medial axis of $P$ in *expected* linear time [12]. Their algorithm adds edges to close of all histogram polygons and applies randomization twice: once to compute the medial axis of all edges of a histogram polygon, and again when non-$P$ edges are removed and the medial axes are merged. The first is strongly predicated on the fact that all edges affect the medial axis, so it is unclear how to directly make it deterministic. Addition and deletion of edges also adds to the programming complexity.

## 2 Preliminaries

Let $P$ be a simple polygon with $n$ vertices, $\{p_1, p_2, \ldots, p_n\}$. The boundary $\partial P$ consists of these vertices and the edges (open line segments) between consecutive vertices. We assume that the vertices and edges of $P$ are in general position, which can be simulated by (actual or conceptual) perturbation of the input.

### 2.1 The Medial Axis and Voronoi Diagram of $P$

The *Voronoi diagram* [2, 15] of a set of *sites* is the partition of the plane into connected regions having the same set of closest sites. This partition consists of *Voronoi cells, edges,* and *vertices,* as in Figure 1. The *medial axis* of $P$ is the locus of all centers of circles contained in $P$ that touch $\partial P$ in two or more points. Thus, the medial axis consists of Voronoi vertices and Voronoi edges. The only Voronoi edges that are not part of the medial axis are the bisectors of an edge and an incident vertex. We will, therefore, concentrate on computing the Voronoi diagram $V(P)$ in this abstract, and obtain the medial axis by removing these Voronoi edges.

The *constrained Voronoi diagram* of a set of sites on the boundary of $P$ is the Voronoi diagram in which distance is measured along a shortest path inside $P$. All of our Voronoi diagrams should be considered constrained Voronoi diagrams, even though we typically omit the word "constrained." In our algorithms, the reflex vertices are always sites, so the shortest paths are line segments.

Algorithms to merge Voronoi diagrams have been important since Shamos and Hoey [17]. See also Kirkpatrick [10] or Klein and Lingas [11].

**Lemma 2.1** *Let $Q$ be a polygon that is divided into $Q_1$ and $Q_2$ by a diagonal $e$. Let subsets of vertices and edges $S_1$, $S_2$, and $S = S_1 \cup S_2$ be the sites in $Q_1$, $Q_2$, and $Q$ respectively. Given the Voronoi diagrams of $S_1$ in $Q_1$ and $S_2$ in $Q_2$, one can obtain the Voronoi diagram of $S$ in $Q$ in time proportional to the number of Voronoi edges that intersect $e$ and the number of new edges added.*

### 2.2 Histograms

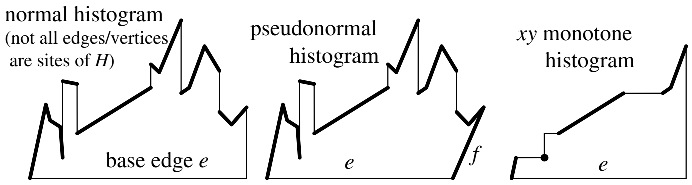


Fig. 2: Histograms

A *normal histogram* (NH) is a simple polygon $H$ whose boundary consists of a *base edge* $e$ and a chain that is monotone with respect to $e$ Typically, we rotate normal histograms so that the base is along the $x$ axis and the rest of the polygon is in the positive quadrant, as in Figure 2. The *base line* is the line through $e$. A *pseudo-normal histogram* (PNH), defined by Klein and Lingas [11], can be viewed as a normal histogram with a missing corner. Because of the merge lemma, an NH is as good as a PNHwith respect to computing a Voronoi diagram.

**Corollary 2.2** *The constrained Voronoi diagram of selected sites of an $n$-vertex* PNH *can be obtained from the diagram of the corresponding* NH *in $O(n)$ time.*

An NH is an *xy-monotone histogram* if, after putting the base along the $x$ axis, the $y$ coordinates of non-base vertices are monotone increasing or monotone decreasing.

### 2.3 Decomposing $P$ into Pseudo-Normal Histograms

Klein and Lingas' algorithm for the constrained Voronoi diagram [11] is based on decomposing a polygon $P$ into PNHs, computing their Voronoi diagrams (via

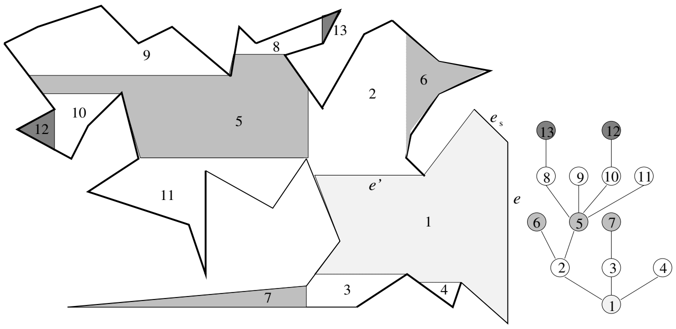

Fig. 3: A decomposition of $P$ into a tree of PNH's

the corresponding NHs), and then merging. In the full paper, we note that only two calls to a linear-time trapezoidation algorithm are needed.

Figure 3 shows a polygon $P$ decomposed into thirteen PNHs. $\mathrm{PNH}_1$ is associated with the vertical base $e$ missing its upper corner; $\mathrm{PNH}_2$ with the horizontal base $e'$ is missing its left corner, etc. The decomposition can be represented as a tree whose nodes are PNHs and whose edges represent adjacency of two PNHs.

**Lemma 2.3** *An $n$-vertex simple polygon $P$ can be decomposed into a set of* PNH*s having a linear number of vertices.*

It has been proved that the Voronoi diagrams of two PNH's do not interfere with each other as long as these two PNH's are (i) at the same depth not facing each other, or (ii) with their corresponding depths more than two apart. Although the proof given by Klein and Lingas is only for vertex sites, the proof can be easily extended to include edge sites.

**Lemma 2.4 (Klein and Lingas [11, Thm 4.6])** *Given the Voronoi diagrams for the* PNH*s in a decomposition of an $n$-vertex simple polygon $P$, the Voronoi diagram for $P$ can be computed in linear time.*

The task that remains is to compute the constrained Voronoi diagram of selected vertex and edge sites inside a normal histogram.

## 3 Computing the Voronoi of an $xy$ Monotone Histogram and the Voronoi Extension

In this section, we provide efficient subroutines for two Voronoi diagram problems. See Figure 4.

**Voronoi of $xy$ monotone histogram** Compute the constrained Voronoi diagram of selected sites in an $xy$ monotone histogram $H$.

**Voronoi extension** Given the intersection of the Voronoi diagram of the sites in a histogram $H$ with the base edge $e$, compute the Voronoi diagram in the halfspace below the base line.

In both problems the sites have an order given by a curve and the intersection of the Voronoi with the curve is known. In this abstract, we focus on the $xy$ monotone histogram $H$, where the curve is the monotone portion of the boundary of $H$. We direct initial and final rays to $x = -\infty$ and $y = \infty$ as shown in figure 4.

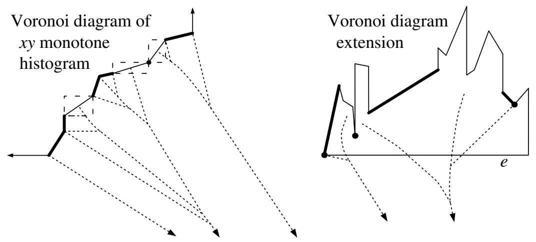


Fig. 4: The diagrams are trees with known leaves

Monotonicity in both $x$ and $y$ not only implies that the boundary of $H$ does not leave the bounding box of the diagonals that join consecutive sites in $x$ coordinate order, but also that the Voronoi diagram inside the box is completely determined by the endpoints of the diagonals. In fact, we replace portions of the boundary that are not sites with these diagonals. To avoid a degenerate case, if an edge and its lower endpoint are both sites, we consider them as a unit.

In the full paper, we sketch a randomized incremental construction that can solve both problems in expected linear time. For deterministic algorithms, we adapt the work of Aggarwal et al. [1].

### 3.1 Deterministic Algorithms

The basic idea of [1] is to identify a fraction of the Voronoi cells that are not adjacent to each other, remove their corresponding sites, recursively compute the Voronoi diagram of the remaining sites, and then independently merge in the non-adjacent cells. Identifying non-adjacent cells is complicated by the fact that one does not have the cell descriptions until the algorithm is done.

Let $s_1, s_2, \ldots, s_k$ be the list of sites in order along the curve. (Include sites at infinity as $s_0$ and $s_{k+1}$.) We mark sites red and blue to satisfy three rules:

1.) No two adjacent sites are *red.*
2.) No three adjacent sites are *blue.*
3.) If the portion of the boundary below 5 sites, ($s_{i-2}$, $s_{i-1}$, $s_i$, $s_{i+1}$, $s_{i+2}$), has a circle below that touches $s_{i-1}$ and $s_{i+1}$ and does not contain any point of $s_{i-2}$, $s_i$, or $s_{i+2}$, then site $s_i$ must be *red.*

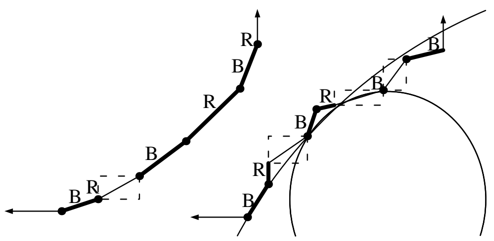


Fig. 5: Possible markings

Figure 5 shows two histograms with possible markings. Recall that if an edge and its lower endpoint are both sites, we consider them as a unit—otherwise symbolic perturbation would be needed so that two open edge sites with the same (non-site) endpoint could not be co-circular with two other sites. Marks on the concave chain at left are constrained only by rules 1 and 2, while the reds at right are forced by the empty circles.

**Lemma 3.1** *A marking of sites can be computed in linear time.*

**Proof:** The first and third rules do not conflict because the third cannot apply to two adjacent sites. Thus, we can mark sites in linear time by initializing all sites to blue, then marking the sites that rule 3 says must be red. For each sequence of $i > 2$ blues that remain, we mark every other other site red, starting with the second and ending one or two before the last. ∎

This coloring has the following independence property:

**Lemma 3.2** *Consecutive red sites cannot have Voronoi cells that are adjacent.*

**Proof:** Consider two sites $s$ and $s'$ that have adjacent Voronoi cells. There is a circle in $H$ that touches $s$ and $s'$ and excludes all other sites.

If $s$ and $s'$ have a single site $t$ between them, then by rule 3, $t$ is marked red and by rule 1, both $s$ and $s'$ are blue. On the other hand, if two sites $t$ and $t'$ lie between $s$ and $s'$, then move the circle center along the bisector of $s$ and $s'$ toward $t$ and $t'$, as in figure 6. The new circle exits the old only between $s$ and $s'$. Therefore, by monotonicity, the circle encounters $t$ or $t'$ before encountering any other. Suppose it encounters $t'$, then $t$ is red by rule 3 and $s$ must be blue. Therefore, either $s$ and $s'$ are not both red or they are not consecutive. ∎

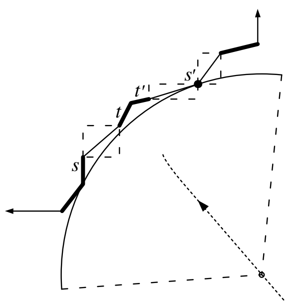

Fig. 6: $s$ or $s'$ is blue

These two lemmas are sufficient to use the algorithm of Aggarwal et al. [1], as we shall briefly describe. They prove the following combinatorial lemma.

**Lemma 3.3 (Aggarwal et al. [1])** *Let $T$ be a binary tree embedded in the plane. Each leaf of $T$ has an associated "neighborhood," which is a connected subtree rooted at that leaf, and leaves adjacent in the topological order around the tree have disjoint neighborhoods. Then there are a fixed fraction of the leaves with disjoint, constant-size neighborhoods, and such leaves can be found in linear time (assuming that neighborhoods can be traced out in breadth-first order).*

The rest follows Aggarwal et al. [1]; see their paper for more detail.

**Theorem 3.4** *The Voronoi diagram of an $xy$ monotone histogram can be computed in linear time.*

**Proof:** By rule 2, a constant fraction of the sites are marked blue. We compute their Voronoi diagram recursively and let $T$ be the tree of Voronoi edges that start between blues sites that are separated by reds. Now, the "neighborhood" of a leaf is the portion of the Voronoi edge that is farther from the blue sites that define it than from the red site that is being inserted. Lemma 3.2 says that adjacent neighborhoods are disjoint, so Lemma 3.3 says that a constant fraction of the red sites with disjoint, constant-size neighborhoods can be found. These red sites can be merged into the blue diagram in constant time apiece.

Finally, a constant fraction of the sites remain red; we again compute their Voronoi diagram recursively and merge it into the blue Voronoi diagram—we can do this in linear total time if we merge connected portions starting and ending on the histogram boundary. ■

We can deterministically compute the Voronoi diagram extension by a similar algorithm.

**Theorem 3.5** *Given the intersection Voronoi diagram of a histogram with the base edge, the extension below the base can be computed in linear time.*

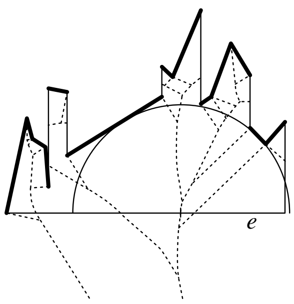


Fig. 7: The constrained Voronoi of selected sites

## 4 Finding the Voronoi Diagram of a Normal Histogram

The key property of the histogram decomposition of Klein and Lingas is that the influence of a site is limited—the Voronoi cell of site does not extend beyond parents, children, or siblings of histograms containing the site. Following Wang and Chin [19], we show that limiting influence is also key to computing the Voronoi diagram of an NH.

Let $H$ be an NH with horizontal base edge $e$. We can identify the sites of $H$ whose influence extends across the base edge by considering circles centered at the base edge that are empty of sites.

**Lemma 4.1** *In an* NH $H$*, the sites whose Voronoi cells extend below the base line are those that can be touched by a circle centered at the base edge and empty of other sites.*

We assume that $H$ has been decomposed into a linear number of horizontal trapezoids, which is the result of our decomposition into PNHs or of running Chazelle's algorithm [4]. We call the horizontal segments introduced by trapezoidation *chords*. The dual graph of the trapezoidation of $H$ is a tree rooted below the base edge whose edges correspond to chords, as illustrated in Figure 8.

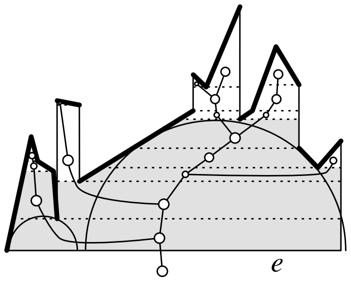


Fig. 8: Trapezoids in an IH

The *influence region* of $H$ is the union of all circles centered at the base edge whose interiors do not intersect a site. The *influence histogram*, IH, consists of all horizontal trapezoids that intersect the influence region.[1] We find the IH by exploring dual tree of the trapezoidation and maintaining stacks of sites whose Voronoi cells may intersect the base.

[1]This influence region and influence histogram are slightly different from Wang and Chin's corresponding ones for the constrained Voronoi diagram of vertices of $P$ [19].

**Lemma 4.2** *The influence histogram of $H$ can be computed in time proportional to the number of its trapezoids.*

**Proof:** All circles in this algorithm will be centered on the base edge $e$. Any two such circles intersect in at most one point above the base line. Let $C_s$ denote the largest circle that crosses a chord $s$ and whose interior does not contain a site on or below $s$. If $C_s$ exists, it either touches sites to the left and right of its center, or is centered at a (non-site) endpoint of $e$, or is infinite. The first case corresponds to the intersection of $e$ with the bisector of the two sites touched, as can be seen in Figure 7.

We compute all such largest circles. Initialize empty stacks $L$ and $R$, for left and right. If the left endpoint of the base edge is a site, insert it into $L$; similarly, insert the right endpoint into $R$. Let chord $s = e$.

Now, we maintain two invariants:

1. With respect to the histogram consisting of the trapezoids output so far, the stacks contain the sites whose Voronoi cells intersect $e$ in order.
2. The largest circle $C_s$ is determined by the sites at the tops of $L$ and $R$.

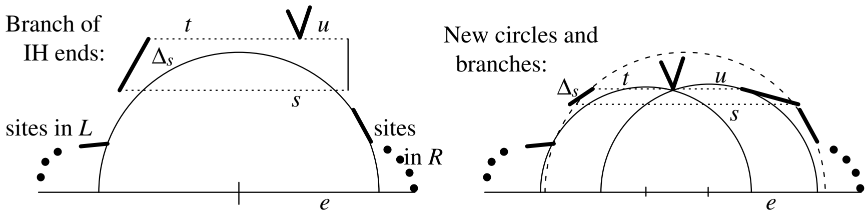


Fig. 9: The largest circle $C_s$ and the trapezoid $\Delta_s$

Let $\Delta_s$ be the trapezoid above $s$ and let $t$ and $u$ be the at most two other chords of $\Delta_s$ as illustrated in Figure 9. Include $\Delta_s$ in the output. If the segment of $C_s$ above $s$ is contained in $\Delta_s$ then this branch of the IH is complete. If $C_s$ crosses chord $t$ but does not contain a site in $\Delta_s$ then let $s = t$ and continue. Otherwise, we need to re-establish the invariants, as at the right of Figure 9.

First, the new sites in $\Delta_s$ may be closer to $e$ than sites in $L$ or $R$—new Voronoi cells may crowd out old ones. If the largest circle determined by the next-to-top site in $L$ and sites in $\Delta_s$ does not contain the top site in $L$, then pop the top of $L$. The first site to remain on the stack certifies that all sites beneath also remain, since stacked sites appear in the same order as their Voronoi cells intersect $e$. Handle $R$ similarly.

Next, we look at the new empty circles crossing chords $t$ and $u$—if there are none, then the branch of the IH ends here. If only one chord is crossed, then we push the new sites whose Voronoi cells intersect $e$ on the $L$ and/or $R$ stacks and continue. Otherwise, both chords are crossed, as in Figure 9; some site, call it $f$, touches $C_t$ on the right and $C_u$ on the left. We continue building the

IH across $t$ using circle $C_t$ and stack $L$ and a new stack $R'$ that contains site $f$ alone. Building IH across $u$ uses $C_u$, $R$, and a new $L'$ containing only $f$. Site $f$ will not be popped off these stacks because among the empty circles between $C_t$ and $C_u$ there must be one that touches $f$ directly above the center—this circle, tangent to $t$ and $u$, certifies that the Voronoi cell of $f$ in histogram $H$ intersects the base edge $e$. Thus, the branches can be computed independently and the invariants can be maintained. ∎

The IH contains all sites whose Voronoi cells intersect the base edge. It may also contain other sites, but they can be grouped into $xy$ monotone histograms.

**Lemma 4.3** *The Voronoi diagram of an influence histogram $H$ can be computed in time proportional to the number of vertices in $H$.*

**Proof:** Sketch: Each time the dual of the IH branches—and a circle is replaced by two—we know a vertical segment to the base edge contained in the Voronoi cell of the site forcing the branch. Cut the histogram along this segment. We are left with pairs, a histogram and a circle that intersect all of its horizontal trapezoids. Thus, the histograms are *bitonic* and can be cut into two $xy$ monotone histograms. Compute the Voronoi diagrams of a bitonic histogram by merging the diagrams of its $xy$ monotone histograms. Merge bitonic histograms by simply rejoining along the vertical cuts. ∎

If we decompose an NH, $H$, into a tree of influence histograms, we can compute its Voronoi diagram in linear time.

**Theorem 4.4** *The Voronoi diagram of selected sites in a normal histogram $H$ can be computed time proportional to the number of vertices in $H$.*

From this theorem, with the lemmas from Section 2.3, we obtain the final result.

**Corollary 4.5** *The Voronoi diagram of the vertices and edges of a polygon $P$ can be computed time proportional to the number of vertices in $P$.*

## 5 Conclusion

We have given an optimal linear-time algorithm for computing the Voronoi diagram or medial axis of a simple polygon.Several problems for simple polygons can be solved in linear time based on this result: computing the largest inscribed circle, building a query structure for the closest boundary point, and determining the buffer zone of all points within $\varepsilon$ of a simple polygonal curve. This algorithm also applies to other $L_p$ metrics and constant-complexity convex distance functions.


## Acknowledgements

We thank Otfried Schwartzkopf and David Kirkpatrick for discussions on medial axis algorithms, and Bethany Chan, Siu-Wing Cheng, and Michael McAllister for their comments on drafts of this paper.

## References


[1] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Disc. & Comp. Geom.*, 4:591–604, 1989.

[2] F. Aurenhammer. Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Comp. Surveys*, 23(3):345–405, 1991.

[3] H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Proc. Symp. Models for Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[4] B. Chazelle. Triangulating a simple polygon in linear time. *Disc. & Comp. Geom.*, 6:485–524, 1991.

[5] O. Devillers. Randomization yields simple $O(n \log^* n)$ algorithms for difficult $\Omega(n)$ problems. *Int. J. Comp. Geom. App.*, 2(1):97–111, 1992.

[6] H. Djidjev and A. Lingas. On computing the Voronoi diagram for restricted planar figures. In *WADS '91: Second Workshop on Data Structures and Algorithms*, number 519 in LNCS, pages 54–64. Springer-Verlag, 1991.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[8] H. N. Gürsoy and N. M. Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part I algorithm. *Engineering with Computers*, 8:121–137, 1992.

[9] M. Held. *On the Computational Geometry of Pocket Machining.* Number 500 in LNCS. Springer-Verlag, 1991.

[10] D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proc. 18th FOCS*, pages 162–170, 1977.

[11] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. In *Proc. 9th Ann. ACM Symp. Comp. Geom.*, pages 124–132, 1993.

[12] R. Klein and A. Lingas. Fast skeleton construction. To appear, *Proc. 3rd Europ. Symp. on Alg. (ESA'95)*, 1995.

[13] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pat. Anal. Mach. Int.*, PAMI-4(4):363–369, 1982.

[14] S. Lu, H. Xu, and C. Wang. Detecting and eliminating false strokes in skeletons by geometric analysis. In *SPIE's OE/Technology'92, Vision Geometry*, Boston, Nov. 1992.

[15] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, 1992.

[16] A. Rosenfeld. Axial representation of shape. *Comp. Vis. Graph. Image Proc.*, 33:156–173, 1986.

[17] M. I. Shamos and D. Hoey. Closest point problems. In *Proc. 16th FOCS*, pages 151–162, 1975.

[18] P. Vermeer. Two-dimensional mat to boundary conversion. In *Proc. 2nd Symp. Solid Model. Appl.*, pages 493–494, 1993.

[19] C. A. Wang and F. Chin. Finding the constrained Delaunay triangulation and constrained Voronoi diagram of a simple polygon in linear time. To appear, *Proc. 3rd Europ. Symp. on Alg. (ESA'95)*, 1995.