Algorithmica (2001) 30: 144–163
DOI: 10.1007/s00453-001-0014-x

Algorithmica

A One-Step Crust and Skeleton Extraction Algorithm[1]

C. Gold[2] and J. Snoeyink[3]

Abstract. We wish to extract the topology from scanned maps. In previous work [GNY] this was done by extracting a skeleton from the Voronoi diagram, but this required vertex labelling and was only useable for polygon maps. We wished to take the crust algorithm of Amenta et al. [ABE] and modify it to extract the skeleton from unlabelled vertices. We find that by reducing the algorithm to a local test on the original Voronoi diagram we may extract both a crust and a skeleton simultaneously, using a variant of the Quad-Edge structure of [GS]. We show that this crust has the properties of the original, and that the resulting skeleton has many practical uses. We illustrate the usefulness of the combined diagram with various applications.

Key Words. Curve reconstruction, Medial axis, Voronoi diagram, Planar subdivision, Scanned maps, Topology building in GIS.

1. Introduction and Previous Work.

Workers in both Computational Geometry and Geomatics have discussed appropriate data structures for managing two-dimensional maps, for example, [B2], [F], [CDW], [H], [NME], [PC], and [W]. However, it is rarely noted that data input is the largest expense in Geographic Information Systems (GIS), and that the spatial data structure has a large influence on the cost of topological data input and structuring. See [G2], [G3], and [GRR] for a discussion of traditional spaghetti digitizing, and the advantages of a space-covering tiling—e.g. the Voronoi diagram—for the creation of topologically connected maps. The book by Okabe et al. [OBS] summarizes many Voronoi applications.

In current practice, the creation of the correct topological connectivity is of considerably greater difficulty than the mere acquisition of the appropriate x–y coordinates. In many cases arcs are digitized by hand, and the polygon construction from these arcs involves many complexities of node determination, arc ordering around nodes, etc. (For a good discussion of current practice see [P]). Gold et al. [GNY] discussed simplifying the polygon mapping process with the introduction of simple Voronoi diagrams, and Gold [G5] extended this to the case where the original paper documents were of sufficient quality that topologically correct maps could be generated directly from scanned input.

[1] Support for this research was received from the Natural Sciences and Engineering Research Council of Canada and the Networks of Centres of Excellence Program, under the IRIS network, and from the Industrial Research Chair in Geomatics at Laval University.
[2] Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, and Department of Geomatics, Laval University, Quebec City, Quebec, Canada.
[3] Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599-3125, USA, and Department of Computer Science, University of British Columbia, Vancouver, BC, Canada.

Received January 17, 1999; revised December 1, 1999. Communicated by M. van Kreveld.
Online publication March 26, 2001.

Fig. 1. A forest map.

Fig. 2. Fringe points.

The Voronoi diagram is implemented using the Quad-Edge structure and an incremental Voronoi algorithm that closely follows that given in [GS] among others. In the rapid digitizing procedure of [GNY], the operator digitizes the interior of each polygon, inserting points around the perimeter, each with the polygon label. The Delaunay triangulation/Voronoi diagram is then constructed, and the triangulation is traversed once, using the visibility order described in [GNY] to extract the Voronoi boundaries between points with differing labels. The result is a set of arcs (polygon boundaries) with guaranteed topological connectedness.

The approach involves the generation of labelled "fringe" points just inside each of the polygon boundaries, generating the Voronoi diagram, and extracting only those Voronoi edges that separated fringe points with differing labels (and thus which come from different polygons). The "skeleton" of the black-line boundaries thus forms the topologically connected map. Figures 1–4 illustrate this process—the small "dumbbells" represent the Quad-Edge/Quad-Arc topology. The difficulties with this approach are that the fringe points have to be labelled, and that only polygon-type maps can be produced.

These limitations are overcome due to the work of Amenta et al. [ABE], who showed that the "crust" of a curve or polygon boundary can be extracted from unstructured

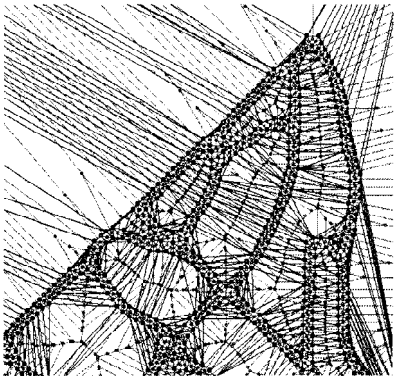


Fig. 3. Voronoi/Delaunay tessellation.

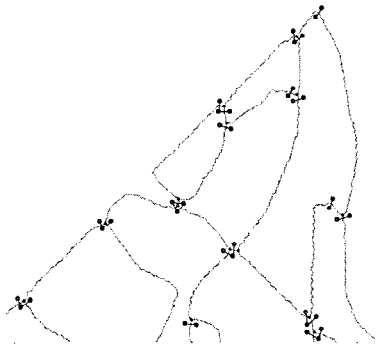


Fig. 4. Extracted arcs.

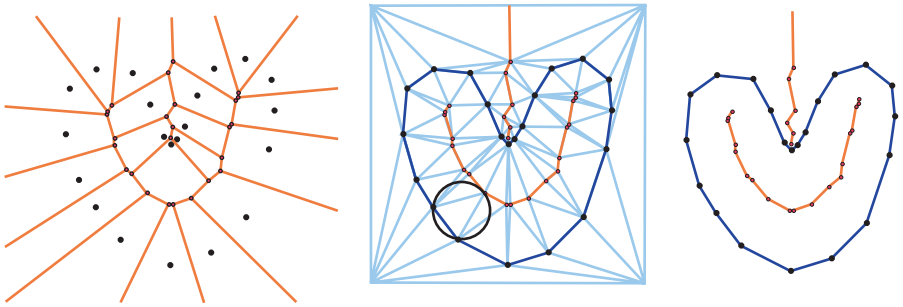


Fig. 5. Crust computation from [ABE]: Compute the Voronoi diagram of sample points, add the Voronoi vertices and compute the Delaunay triangulation, and keep edges joining sample points. Voronoi edges in the sample that do not correspond to crust edges form a skeleton.

(and unlabelled) input data points if the original curve is sufficiently well sampled. Their intuition was that, as the vertices of the Voronoi diagram approximate the medial axis (or skeleton) of a set of sample points from a smooth curve (Figure 5, after [ABE]) then by inserting the original vertices plus the Voronoi vertices into a Delaunay triangulation the circumcircles of this new triangulation approximate empty circles between the original smooth curve and its medial axis. Thus any Delaunay edge connecting a pair of the original sample points forms a portion of the sampled curve—the "crust." In subsequent papers [AB], [ABK] they extended this to three dimensions, extracting triangulations of the surface based purely on the x, y, z coordinates of surface sample points.

This solves admirably the problem of extracting the crust—but in our work on scanned maps we wished to extract the skeleton between our rows of fringe points. Experimentation with the crust algorithm showed that, while crust edges (connecting pairs of original sample points) are extracted correctly, connecting pairs of Voronoi vertices do not necessarily produce a good approximation to the skeleton, as in the construction of the second Delaunay triangulation additional edges are added between Voronoi vertices—see Figures 6 and 7. Our objective was to extract both the crust and the skeleton, in order to process various types of map input.

2. The One-Step Algorithm. Further consideration led to the examination of the relationship between Voronoi edges and Delaunay edges in the original Voronoi/Delaunay construction. This was made simpler by the use of the Quad-Edge data structure [GS], where two of the pointers refer to Delaunay vertices, and two of the dual Voronoi vertices. Our intuition was to apply the crust test to individual Quad-Edges on the original diagram, rather than creating a second structure. The circle in Figure 5, for example, contains a crust edge but no skeleton edges. Thus, for each Quad-Edge, we wished to determine if the Delaunay edge had a circle that was empty of Voronoi vertices (and hence of a portion of the skeleton), and which intersected the Delaunay edge.

Each Delaunay edge is adjacent to two triangles whose circumcircles are centered at Voronoi vertices. In fact, the Voronoi edge between these circle centers is the dual to the Delaunay edge. The original crust essentially tests each Delaunay edge to make sure

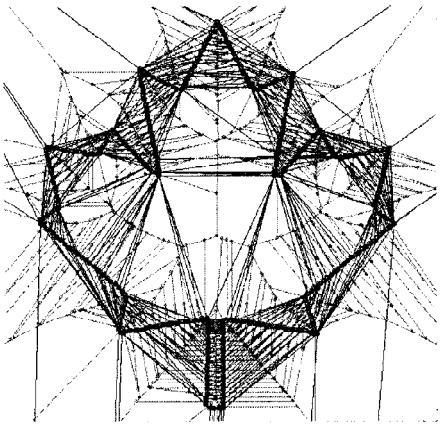


Fig. 6. Maple leaf.

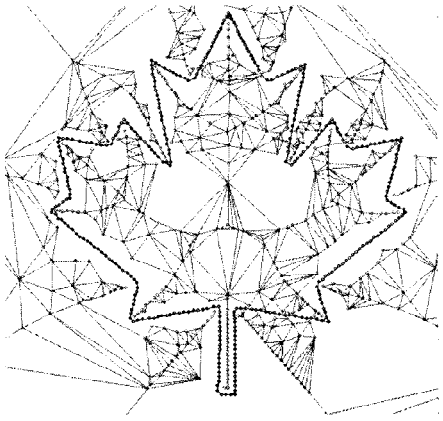


Fig. 7. Crust and residual edges—the two-step approach.

that it has a circle that contains the edge, but does not contain any Voronoi vertices. We make this a local test—testing only the two Voronoi vertices that are the endpoints of the dual Voronoi edge. For a Delaunay edge (q, r) with dual Voronoi edge (a, b) the test is the standard InCircle test [GS] on these four points. *InCircle*(q, r, a, b) determines whether point b is outside the circle through (q, r, a), assuming that this circle is oriented counterclockwise. If so, then that circle shows that edge (q, r) should be included in the crust. On the other hand, if b is inside, then every circle containing (q, r) also contains a or b, and (q, r) is not included in the crust. If the circle through (q, r, a) is oriented clockwise, then the sense of the test is switched. This does not need to be handled as a special case, however, because when it occurs the Voronoi edge (a, b) does not intersect the Delaunay edge (q, r). The same test will correctly indicate that (q, r) should not be included in the crust. The standard *InCircle*(q, r, a, b) test evaluates the sign of a four by four determinant whose row for each point p is $[1, px, py, px^* px + py^* py]$. An alternate expression of this test is useful if the original data are the Delaunay vertices or sites, and the Voronoi vertex coordinates are derived. Suppose that (p, q, r) and (r, q, s) are the two triangles incident on Delaunay edge (q, r) and let v be the vector 90° clockwise from $(r - q)$. Then the test

$$(s - q) \cdot (s - r)^*(p - q) \cdot (p - r) \geqq -(s - r) \cdot v^*(p - q) \cdot v$$

will be true if and only if the edge (q, r) should be in the crust. This shows that the test is a polynomial of degree 4 in the input variables. Thus, this local test can be evaluated exactly with less precision than the global test, which requires evaluation of polynomials of degree 12 in the input coordinates.

A second idea was that Quad-Edges that failed the crust criterion were part of the skeleton or "anti-crust." This term was mentioned briefly in the conclusions of [ABE], citing [RCGH], [BA], and [O]. This is based on the idea that the dual of a crust edge is a Voronoi edge that intersects the crust—and has been rejected. The remaining Voronoi edges form a "tree" structure that extends towards the crust but does not cross it. (Indeed,

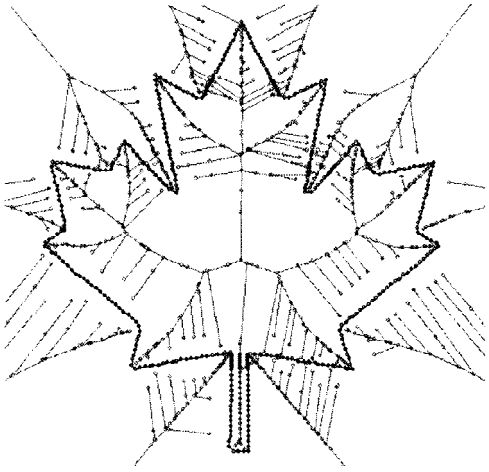


Fig. 8. One-step crust and skeleton.

with the Quad-Edge structure, each leaf of the skeleton is associated with a particular crust vertex.) The one-step algorithm consists of *assigning* Quad-Edges either to the "crust" or the skeleton ("anti-crust"), instead of constructing a second diagram, as shown in Figure 5. The results for Figure 6 are shown in Figures 8 and 9. Mis-assignments occur where the sampling conditions of [ABE] are not met—especially at acute angles. We later learned that Attali had previously given an angle condition on Delaunay circles that is equivalent to the local test [A]. Our expression is somewhat easier to compute, but gives the identical graph.

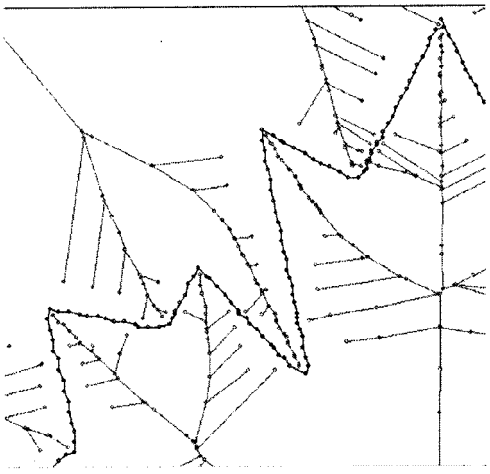


Fig. 9. Enlargement of part of Figure 11.

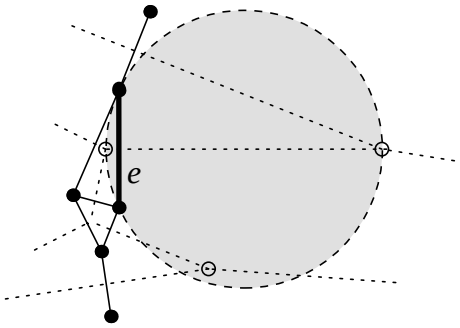


Fig. 10. Edge e is in the locally defined crust, but not in the globally defined crust.

The leaf vertices or "hairs" on this skeleton exist where there are three adjacent sample points whose circumcircle contains no other sample point—at the end of a major branch of the skeleton, or at a minor perturbation of the sample points. These reach out to every minimum of curvature [AS]. For a true curve they would only occur at the end of major branches, as the skeleton is formed wherever a circle can touch two (not three) points on the curve.

3. Relationship of the Local "One-Step" Algorithm to the Crust Algorithm. Figure 10 illustrates that selecting crust edges based on a local property instead of a global property may allow additional edges to be added to the crust. In this figure, the heavy black edge e is in the locally defined crust because there are circles that contain e but do not contain the endpoints of the Voronoi edge that is dual to e. On the other hand, e is not in the globally defined crust because every circle that contains e does include some Voronoi vertex. We can observe, however, that when the sampling condition of Amenta et al. [ABE] is satisfied, then the locally defined crust is identical to the globally defined crust. In addition, we improve their analysis and relax the sampling conditions.

Amenta et al. define the *Local Feature Size*, $LFS(p)$, for a point p on a smooth curve to be the distance from p to a closest point on the medial axis of the curve. A curve is *r-sampled*, for some $0 < r < 1$, if, for every point p on the curve, there is a sample point within $r^* LFS(p)$ of p. They prove two theorems for the globally defined crust of an r-sampled curve: First, when $r < 0.40$, every edge between adjacent samples is included in the crust. Second, when $r < 0.252$, no extra edges are included.

The first theorem immediately applies to the locally defined crust: the local condition is less restrictive than the global, so that any edge that passes the global test will pass the local test. We improve the bound to $r < 0.5$ for the globally defined crust and to $r < 0.62$ for the locally defined crust (neither of which are best possible). We then show that the proof of the second theorem applies to the locally defined crust, and improve the bound to $r < 0.42$.

We use relationships between local feature size, Voronoi circles, and the curvature that extend a lemma of Amenta et al. [ABE].

LEMMA 1 [ABE]. *For any point p on a smooth curve, the two circles tangent to p of radius $LFS(p)$ contain no points of the curve.*

LEMMA 2. *Let s, t and u be points on a smooth curve* (*or curves*), *and let v be the center of their circumcircle. There is a point of the medial axis in the convex hull of s, t, u, and v.*

PROOF. Let p be the point on the curve closest to v. We may assume that p is unique; otherwise, v is on the medial axis and we are done. Note that this assumption implies that p is distinct from s, t, and u. We may further assume, without loss of generality, that if we delete points s and t, then points p and u would lie on separate components of the curve. Figure 11 illustrates a simple situation.

Now, consider moving a point v' from v to u; its closest point p' on the curve is initially p and finally u. Closest point p' cannot travel continuously, however, because the distance $d(v', u)$ is always less than both $d(v', s)$ and $d(v', t)$. Therefore, some point v' will have more than one closest point on the curve and will be on the medial axis. □

Now we can establish bounds on r such that all desired edges are present in the crust.

THEOREM 1. *The crust of an r-sampled smooth curve contains an edge between every pair of sample points that are adjacent on the curve, for $r < 0.5$. For the locally defined crust, this can be improved to $r < 0.62$.*

PROOF. Consider a pair s, t of adjacent sample points and let p be the point on the curve between them that is equidistant from s and t. Assume that $LFS(p) = 1$; this means that there is a unit circle B' around p that contains no point of the medial axis of the curve. By Lemma 1, it also means that the two circles C and C' of unit radius that are tangent to the curve at p have no other intersections with the curve. These are illustrated in Figure 12.

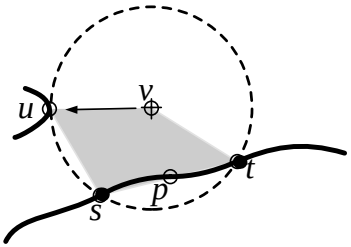


Fig. 11. We find a point of the medial axis along vu.

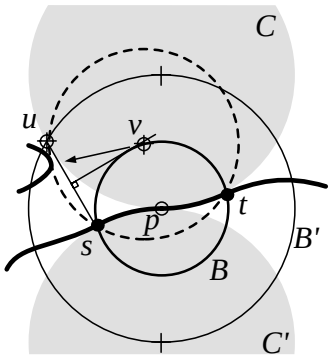


Fig. 12. The curve remains outside of the circles C and C' tangent to p, so if there is a Voronoi vertex v in circle B, then there is a point of the medial axis in circle B'.

Suppose, for the sake of deriving a contradiction, that some vertex v of the Voronoi diagram lies inside the circle B of radius r around p. If we can show that the Voronoi circle for v remains inside the union $B' \cup C$, then we will be done—since samples cannot lie in C, Voronoi vertex v, and the three samples that define v, will all lie in B', and Lemma 2 then reveals a point of the medial axis inside their convex hull.

We use the fact that both s and v lie inside B to show that the Voronoi circle V remains inside the union $B' \cup C$ when $r < 0.5$. For a fixed center v, point s can be moved downward to increase the distance $d(s, v)$ until s is at the intersection of circles B and C'. The closest point to v of the boundary of $B' \cup C$ is the point u at the intersection of two circles. The quantity $d = d(s, u)/2$ can be calculated by noting that $s = ((r^4/4 - r^2)^{-2}, -r^2/2)$ and $u = (-\sqrt{\frac{3}{2}}, \frac{1}{2})$. The circle V first touches the perpendicular bisector of su when $1 - (r + d)^2 = r^2 - (r - d)^2$, and is below for all values of $r < 0.50720648$.

For the locally defined crust, we can use ideas from the proof of Lemma 2 to improve the bound to $r < 0.62$. Suppose that B contains a Voronoi vertex v defined by adjacent sample points, s and t. We allow the third sample point u to lie outside of B', but when $r < 0.62$ calculations like those of the previous paragraph show that the point u will be within $(2 - r)$ of the center p.

We argue that when u is within $(2 - r)$ of the center p, then there is still a point of the medial axis inside B'. First, if B' (or any smaller circle centered at p) intersects the curve in three points, we could choose those points as Voronoi sites to make p a Voronoi vertex, then apply Lemma 2 to reveal a medial axis point inside B'. Thus, we may assume that v has its closest point on the curve between s and t and inside B. When we walk from v to u maintaining the closest point, it must switch outside of B before we are within $(1 - r)$ of point u, thus the medial axis point is inside B'. □

When we wish to guarantee that no undesired edges are present in the crust, then the bounds on r are necessarily smaller. Figure 13 shows a dumbbell-shaped example in which two non-adjacent points are joined by a crust edge when $r > 0.48653$. In Theorem 2 we sketch a proof that this never occurs when $r < 0.42$. We take a few moments to describe the example, because it is the worst configuration that arises in the proof of Theorem 2.

Draw a unit circle B centered at the origin, and choose points s and t where B intersects the y axis. Draw two Voronoi circles through points s and t that are centered where B intersects the x axis—their radii will be $\sqrt{2}$. Draw two circles of radius ρ that are centered on the y axis at $x = +/- (1 + \rho)$, where ρ is the positive root of $\rho^4 + 2\rho^3 - 2\rho - 2$ whose value is about 1.1069. Choose the four points (other than s and t) where these circles intersect the Voronoi circles as the samples adjacent to s and t, and draw the arc between them; the points adjacent to s are labeled u and u' in Figure 13. Finally, complete the curve by arcs of two circles centered on the x axis that are tangent to these arcs, and sample densely on these portions of the curve.

Consider the point p that is the midpoint of the arc between s and u. The choice of the magic number ρ causes point p to be equidistant from the medial axis inside and outside the curve and, in fact, $LFS(p) = \rho$. The ratio $\mathrm{d}(s, p)/LFS(p)$, which determines r, turns out to be $\sqrt{(2\rho^2 - 2\rho)}$, which is less than 0.48653.

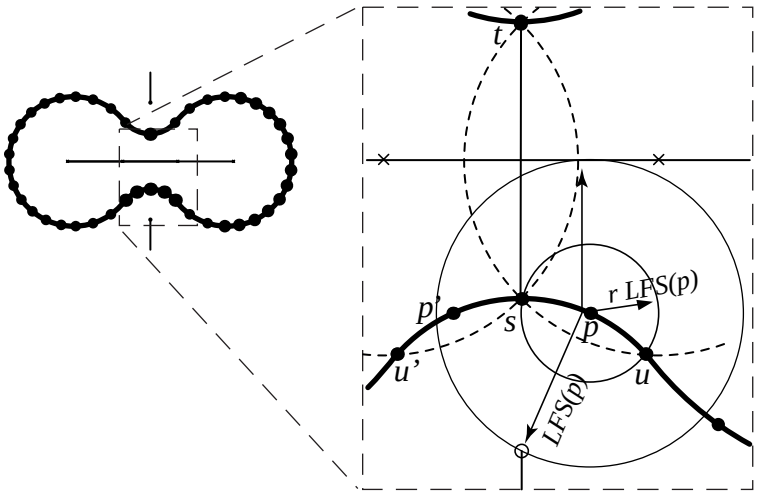


Fig. 13. Non-adjacent points s and t in the pinched part of this curve will be joined by an edge of the crust when $r > 0.48653$.

In the next theorem we establish bounds on r for which we can guarantee that no undesired edges are present in the crust.

THEOREM 2. *The locally or globally defined crust of an r-sampled smooth curve does not contain an edge between non-adjacent sample points, for $r < 0.42$.*

PROOF. The idea is to show that for non-adjacent sample points s and t to be joined by an edge, we must have a point p on the curve midway between s and an adjacent point u that gives a lower bound $r \geq \mathrm{d}(s, p)/LFS(p)$, and that is lower bound is minimized essentially by the configuration of Figure 13. In that example, we knew $LFS(p)$ exactly because we knew the entire medial axis; in this proof, we upper bound $LFS(p)$ by just knowing a couple of places where we must find points on the medial axis. Thus, we sketch a proof that, when $r > 0.4208614 > \sqrt{\left(3+\sqrt{\frac{7}{2}}\right)}$, non-adjacent points s and t are not joined. This is best illustrated with a system like Cabrii or Sketchpad.

Let s and t be non-adjacent sample points. Assume, for the sake of deriving a contradiction, that there is a circle B that touches s and t, is empty of sample points, and is crossed by the Voronoi edge that is the portion of the bisector of s and t.

Let v and v' be the points in which this Voronoi edge intersect the circle B; the circles V and V' through s with centers v and v', respectively, are empty of sample points. Point s has two adjacent sample points, u and u', which must therefore be outside of V and V'. Let p be the point on the curve equidistant from s and u, and let p' be equidistant from s and u'.

Since u and u' are outside the circle B, a point moving from s to t along st cannot have its closest point on the curve change continuously. Thus, st intersects the medial axis.

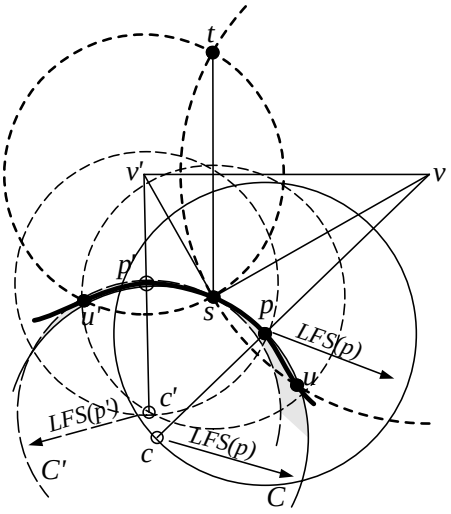


Fig. 14. The general configuration in Theorem 2.

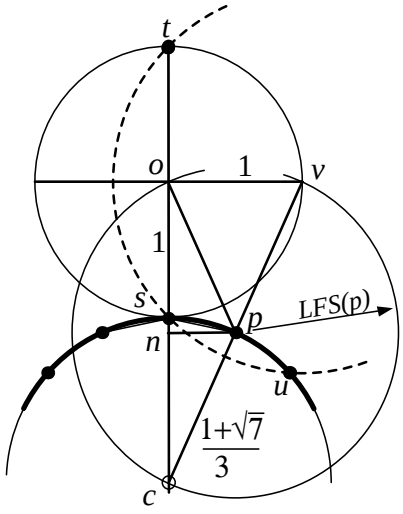


Fig. 15. The configuration defining the bound for Theorem 2.

We may assume that s and t are chosen such that the point of intersection is closer to endpoint s, and thus one upper bound on $LFS(p)$ is the distance from p to the midpoint of st.

Consider the circle C of radius $LFS(p)$, passing through s and u, that is centered at a point c below the curve, as in Figure 14. Circle C must contain p, and must not intersect the curve between s and u', or else Lemma 2 would reveal a point of the medial axis within $LFS(p)$ of p. Similarly, circle C' of radius $LFS(p')$ through s and u' with center at c' contains p' and not p. Note that this implies that c is right of the line $c'p'$, and c' is right of the line pc.

If $LFS(p') \leq LFS(p)$, then we can move c' away from p' along the ray $v'p'$ and decrease the bound $\mathrm{d}(s, p')/LFS(p')$ until either circle C' hits p, or $LFS(p') = LFS(p)$. In the former case, there is a point of the medial axis within $LFS(p)$ of p. In the latter case, we can move c and c' until they meet on the circle of radius $LFS(p)$ centered at s, and the distances $\mathrm{d}(s, p) = \mathrm{d}(s, p')$. This occurs when line cs bisects the angle $\langle vcv'$.

Now, further calculations show that the bound on the ratio $\mathrm{d}(s, p)/LFS(p)$ is minimized when st bisects vv', which places c on the line st in the configuration of Figure 15. Let o be the origin and let n be the midpoint of co. Since cpo is an isosceles triangle, np is perpendicular to co. Triangles cpn and cvo are similar right triangles, so we can calculate that $LFS(p) = (1 + \sqrt{7})/3$ in this configuration, and $\mathrm{d}(s, p) = \sqrt{(5 - \sqrt{7})/3}$. This gives a ratio of $\sqrt{((3 - \sqrt{7})/2)}$, and establishes a lower bound for r of about 0.42806. □

Thus, even under our local definition, the crust of a collection of 0.42-sampled, smooth, closed curves will consist of the collection of exactly those edges of the Delaunay triangulation that join adjacent sample points. To close this section, we would like to justify our continued use of the word *skeleton* to describe the "anti-crust"—the set of

Voronoi edges that are dual to the non-crust edges of the Delaunay. This should not be a surprise, because the medial axis of a planar region is often approximated by the Voronoi diagram of points sampled along its boundary.

In particular, we show that each region bounded by crust edges can be "collapsed within itself" to the skeleton edges within that region. Topologically, the skeleton edges in each region are deformation retracts for the region, with the exception of some extra "hair."

THEOREM 3. *In any set of point sites, each region defined by crust edges has a deformation retraction onto the skeleton in that region, where the skeleton consists of all Voronoi edges whose dual Delaunay edges are not in the crust.*

PROOF. First, we make one observation. For any crust edge e, there is a circle that encloses e and is crossed by e's dual Voronoi edge v. However, then e and v intersect, and the closest site to any point of e is one of the endpoints of e. Therefore, no Voronoi edge other than the dual of e can cross e.

We can easily build a deformation retraction that maps the plane minus the point sites onto the Voronoi diagram—simply map each region away from its closest site. Note that crust edges will be mapped to where they intersect their dual Voronoi edge, and that no point will be mapped across a crust edge. Thus, we can convert this to a retraction that maps regions bounded by crust edges onto the skeleton—simply map each removed Voronoi edge to its two Voronoi vertices on either side of the crust edge. Since an edge can cross the crust at most once, this maps each region bounded by crust edges onto a one-dimensional skeleton, except for triangular regions, which are mapped to the dual Voronoi vertex that they contain. □

This deformation retraction maps regions onto a set of edges that can contain hair and even trees. It is straightforward to change the mapping, sending leaves to parents and deleting leaves, so that the range has no vertices of degree one. Thus, we are justified in calling this a skeleton of the crust—it has the same topological structure as the set of regions.

4. Applications

4.1. *Polygon Generalization.* Following our original objectives, we have processed a variety of polygons and scanned maps with the new algorithm. In the case the of simple maple-leaf polygon of Figure 8, both the crust and the skeleton are extracted essentially correctly. In the case of the crust, sharp corners fail to satisfy the crust sampling criterion and, as expected, an occasional Delaunay edge crosses the tip. In the case of the skeleton, the form is correct but some extraneous branches are generated where perturbations of the boundary data points are treated as incipient salients. This also is a well-known situation. Indeed, the "hairs" on this skeleton form a tool for curve generalization or simplification, as they represent minima of curvature, as described above. Simplification of the skeleton by removing individual hairs is achieved by perturbing or removing

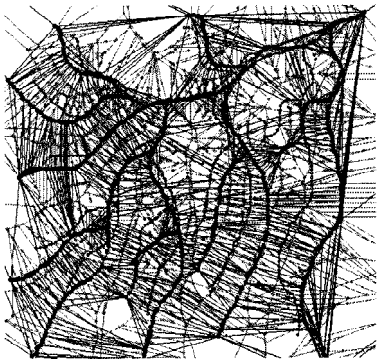


Fig. 16. A river network.

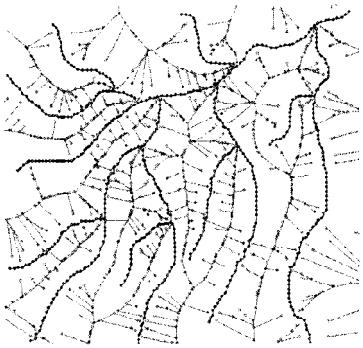


Fig. 17. Estimated watersheds.

individual crust points so as to remove these minima of curvature. The resulting curve has a simple skeleton, and may be generalized in the sense of [OI] or an equivalent.

4.2. *Road or River Network Estimation from Boundaries*. The skeleton of the polygon of Figure 8 forms an interesting approximation of drainage network development in a homogeneous terrain, given the watershed. While obviously not conforming to the non-homogeneous reality, it may well be a useful approximation. An equivalent application is the design of the road network for access to all parts of a homogeneous terrain—perhaps the forest harvesting.

4.3. *Hinterland or Watershed Estimation from Road or River Network*. Figures 16 and 17 show the reverse operation: a river network is represented as a set of sample points, and the one-step algorithm applied. The crust and the skeleton are recognized with the one-step algorithm, and the connected components linked with the algorithm described in [GNY] for rapid digitizing, and in [G5] for scanned maps using the Quad-Edge data structure. The estimated sub-basins may be clearly seen, and extracted, on the basis of the skeleton. This approach has been used successfully in British Columbia for preliminary watershed estimation, with later correction by direct observation. Notice the cases of inadequate sampling according to the [ABE] criterion, especially at intersections, causing the skeleton to break through the river network. Similar results may be obtained if the construction is based on a transportation network. In both cases the catchment area may be estimated for any point on the network, permitting the development of flow or transportation capacity maps. This is possible because of the "hidden" links remaining in these diagrams—the underlying data structure preserves the relationships between skeleton vertices and the associated crust segments, allowing calculation of the increase in catchment area when moving from node to node up the river network.

4.4. *Scanned Maps*. Scanned maps, similar to those shown in the first section, may be processed using the crust criterion instead of vertex coloring. Thus edge pixels for each blank line are extracted, as previously, but they are not labelled by floodfilling polygons. Instead, the crust criterion is applied to extract the crust of each black/white boundary.

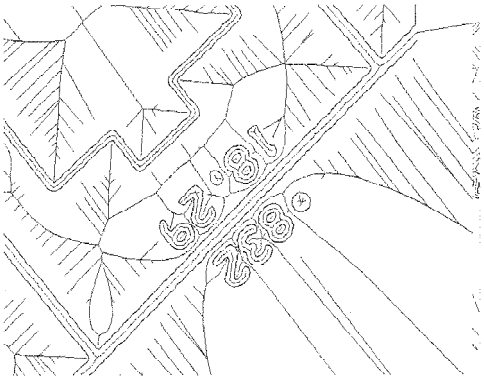


Fig. 18. Crust and skeleton of part of a scanned cadastral map.

As this is purely geometric, it is applicable to closed polygons, connected networks, and unconnected black lines, such as text. These crusts separate the black/white portions of the scanned maps. However, the skeletons are generated by the same algorithm—but for the white portions of the map as well as for the black portions. If the black line work forms a connected graph of a polygon set, or a network, then the skeleton of this black region forms the connected topology representing the line work—as in [GNY] and Figures 1–4. The skeletons of the white regions express the relationships between disconnected black line work, or else express the form of the white shapes, as in [B1]. The crusts may or may not be preserved. As in Figure 18, the crust may break if the sampling criterion of [ABE] is not preserved—especially at sharp junctions. This will cause the "white" skeleton to break through the crust and connect with the "black" skeleton. Several alternatives exist in this case: we may revert to the labelled vertex algorithm; we may improve the sampling; or we may take advantage of the fact that, as all Voronoi vertices fall on one side or the other of the line of the crust, they may each be labelled as "black" or "white" by reference back to the original image. Links are then broken if they connect a black vertex with a white one.

As shown in Figure 18, the skeleton generation algorithm allows for the extraction of the topology of the line work as well as for the detection and placement of text. Burge and Monagan [BM1], [BM2] also worked on the extraction of text from scanned maps, without the topology emphasis. For scanned maps the crust is the boundary between black and white pixels, whereas there is a skeleton for the white regions ("polygons") as well as for the black regions (line work). These last form the centerlines of the scanned line work: skeletonization in the Euclidean, rather than the raster, sense, and form a topologically complete graph of the original input map. It should be noted that the algorithm using labelled fringe points may do better for a simple polygon map, if sampling does not achieve the crust criterion, as more information is supplied about the desired boundary connectivity.

4.5. *Processing Scanned Text*. Scanned text may equally well be treated in the same way as in the scanned map example, after processing with an edge-detection filter, as

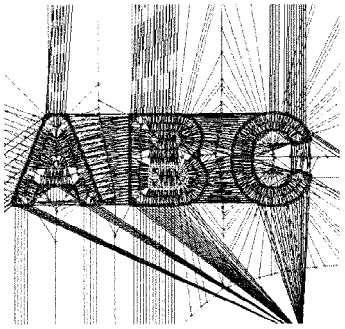


Fig. 19. Character outlines.

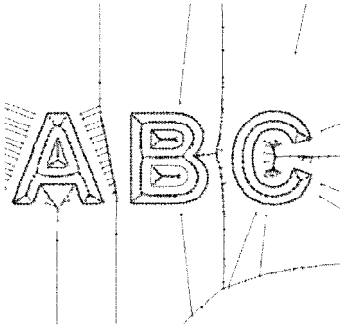


Fig. 20. Interior and exterior skeletons.

shown in Figures 19 and 20. The interior skeleton of the character can detect the characteristic form of the connected graph, essentially for the cost of constructing the Voronoi diagram. The exterior skeleton may be used to express the relationships between the characters. These relationships are also useful for languages using non-connected components, or diacritical marks or accents. Burge and Monagan's methods [BM1], [BM2] involved extracting the text from cadastral maps using Voronoi diagrams—adjacent pixels were given the same label, and the exterior skeletons were used to connect the letters together to form complete words. As in the original work by Blum [B1] the interior skeletons of arbitrary shapes may be used as a shape descriptor.

5. Building the Topological Structure

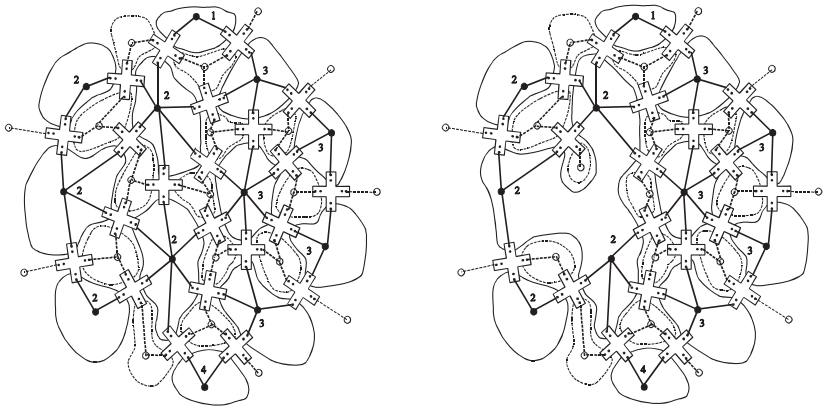
5.1. *Advantages of the Quad-Edge Structure.* So far we have been concerned with the assignment of individual Delaunay or Voronoi edges to the crust or the skeleton. Our original motivation was to generate topologically connected graphs from either or both of these, in order to produce topological maps for GIS-type applications. (This has usually been a very elaborate process.) Thus we needed to extract connected arcs from the individual segments identified by the one-step algorithm, and to leave them connected to each other to form our desired topological map. Traditionally, the construction of a topologically connected network has been the domain of large GISs. The GIS "topological operators" of inclusion, intersection, etc., apply to polygons or other objects that have previously been built, usually by these large systems. The methods described here are operational on PCs, and can generate topological networks, individual polygons, or other forms of output as required.

We needed a structure that was capable of managing both the primal and the dual graph with equal felicity, and that could apply to simple Delaunay/Voronoi structures as well as to polygon arcs. Since we were often working interchangeably with Delaunay/Voronoi representations, the Quad-Edge data structure of [GS] became an obvious candidate. It has various attractions: it is a method for representing the edge connectedness of any connected graph on a manifold; it is symmetric in its storage of both the primal and dual edge; it has a complete algebra associated with it, requiring no searches around polygons, nodes, etc., and it is admirably adapted to an object-oriented programming environment.

There are only two operations: Make-Edge to create a new edge, and Splice to connect or disconnect two ends. Gold [G5] shows a half-page of code that implements the basic operations. To us it appears the most elegant structure. One limitation is that it works only for connected graphs. If there are "islands" in a polygon map, for example, and the arcs between the nodes in the data structure graph correspond to the polygon edges in the map, then the presence of the island within some other polygon cannot be detected without further processing. The Voronoi diagram, being space-covering, does not have this problem.

Since the algebra for the Quad-Edge structure includes operations for moving from arc to adjacent arc, and for obtaining the vertices, we can navigate within this structure in several ways: We can traverse the whole graph using the classic depth-first or breadth-first searches [S], which do not require any geometric information, or using the "Visibility Ordering" of a triangulation [GC], [FFNP], [GNY], [BKO], [SS]. It is also possible to walk from some starting point to some destination point through edges of the triangulation using the "CCW" orientation test on vertices [GCR], [GS] (although it should be noted that this algorithm is only guaranteed for Delaunay triangulations).

5.2. *The Quad-Arc Data Structure*. The original Voronoi structure contains far too many individual points to be used as an archival mechanism, or for internal processing or polygonal maps. It would be desirable as an alternative to remove the large number of irrelevant Voronoi edges without destroying the desired polygonal structure. This would be difficult with an underlying triangulation, but straightforward with the Quad-Edge structure. For our scanned map application, we use a modification of the "Visibility Ordering" traversal implementation of [GNY] to inspect all edges and remove those defined as "crust" rather than "skeleton." We show a small example; this method has been described in [G5]. Figure 21 shows a small triangulation represented as a Quad-Edge data structure and the same structure with two Quad-Edges removed because they were classified as "crust" rather than "skeleton." Figure 22(a) shows the final result of this



(a) (b)

Fig. 21. Initial triangulation of numbered points (a) has two edges removed (b).

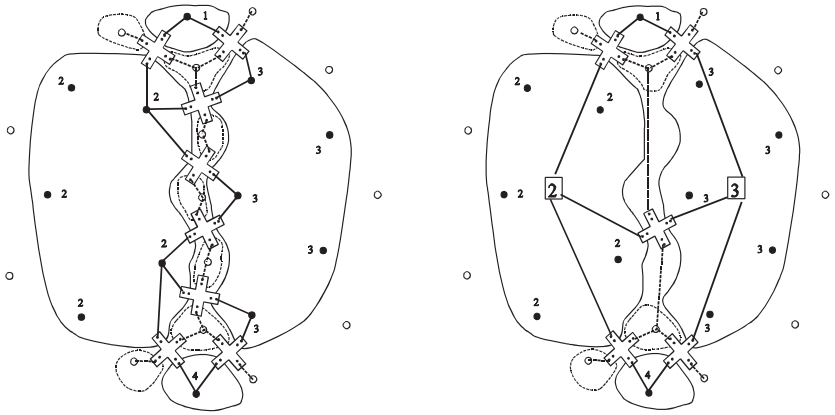


Fig. 22. After removing all edges that join the same numbers (a), one final pass computes the Quad-Arc (b).

process. The graph that remains consists of strings of Quad-Edges ("arcs"), connected at the ends ("nodes"). The graph may be traversed again, using a depth-first search or equivalent, to remove the multiple links while preserving the nodes, as shown in Figure 22(b), so that each of the Quad-Edges indicates an "arc" or polygonal line that joins two nodes rather than indicating a segment. We have called the result of this a Quad-Arc data structure, to emphasize its similarity to the arcs used in a traditional GIS to define the boundary between two polygons. Its topological properties are unchanged from the original Quad-Edge of [GS].

In the original Voronoi structure the positions of islands or other unconnected features within polygons is clearly defined by the adjacency of the Voronoi cells representing the island with those representing the enclosing polygon. The arc extraction process loses this information because all connecting edges are deleted, and the Quad-Edge algebra applies only to connected graphs. By keeping track of the connected components as edges are deleted, however, it is possible to identify the formation of islands and preserve a dummy arc, flagged as such, to keep the link between the island and the containing polygon.

5.3. *Advantages of the Quad-Arc Structure.* Once the structure is completed, the resulting topological structure may be used to perform polygon shading, neighbour detection, etc., as with any mapping program. Due to the overall simplicity of the algorithm, the method may be used in a wide variety of applications. Current work is on a simple mapping package where hand-drawn maps may be scanned into a PC with an economical scanning unit, and the resulting topologically complete map is available in a very few minutes. This may then be exported to a commercial mapping program or GIS, or else preserved internally for simple mapping projects. We have not studied the editing process in great depth yet, as we envisage that it would be simpler to modify the paper map and re-process it.

The original motivation for this work was to take advantage of our scanned mapping

technique. This started with the complete Voronoi structure, which guarantees topological completeness of the processed map, but uses large amounts of storage. Using the Quad-Edge representation of the Voronoi structure permits the elimination of unwanted edges, and the Quad-Edge to Quad-Arc simplification provides even greater storage efficiency. The resulting structure is then equivalent to other polygon-management data structures, in that if there are many intermediate points along the arcs, then the number of pointers contributes little to the storage costs. If there are k points in each resulting arc, then extraction of the Quad-Arcs of the crust would remove $3k$ triangle edges (Quad-Edges), each with 12 pointers in this implementation. One Quad-Edge would replace them, and the points would be saved in a list. This storage would be doubled if the skeleton was preserved as well as the crust, and if there was only a single skeleton branch, as might be the case after generalization with the methods described above. Savings of up to 95% are possible if only simple crusts are preserved.

The resulting data structure cleanly separates the topology from the attributes, in that the pointer structure is entirely within the topological elements: four pointers between the branches of each Quad-Arc in the object-oriented version, four pointers to adjacent Quad-Arcs, and four pointers to attribute information. These are traditionally the two faces and the two nodes associated with the Quad-Edge, but in the Quad-Arc the face pointers point directly to polygon attribute information, perhaps in a database, and the two node pointers may be used to reference the associated arc (chain of x–y coordinates) or other arc information as desired.

Thus only one topology file is required, along with a geometry (arc) file and a polygon attribute storage mechanism. Islands may be integrated with dummy Quad-Arcs. The topology permits any connected graph (not only polygons), thus allowing the maintenance of network graphs. Independent points may be handled in the same way, considering the edges of the Delaunay triangulation to be dummy Quad-Arcs.

Queries in this structure are based on the edge-algebra developed in [GS]. Finding the boundaries of a polygone for example requires successive calls to Oprev (or to Onext to go round in a clockwise direction), until one returns to the starting arc. To find adjacent polygons at the same time involves looking at the dual branches of each Quad-Arc at each step. Since travelling from any Quad-Arc to any other involves only a sequence of "Rot" and "Next" pointers, these may be encoded as a binary string for any path within the map.

The Quad-Edge or Quad-Arc data structure permits topology maintenance in a mapping system with a minimum of complexity. It fits very easily with current scanned map software using the Voronoi model for data input and topological structuring, and permits straightforward topology maintenance and spatial queries after simplification. It is simple to implement in an object-oriented environment, reducing some of the complexity and mystery of cartographic topology.

6. Other Applications. While many other applications of the one-step crust algorithm in mapping come to mind, one more must suffice—but this time without the necessity for Quad-Arc extraction. Another source of map data comes from contour maps, and terrain elevation models are in great demand, especially triangulated elevation models. However, it is not always easy to extract good triangle sets from contours alone—and manually

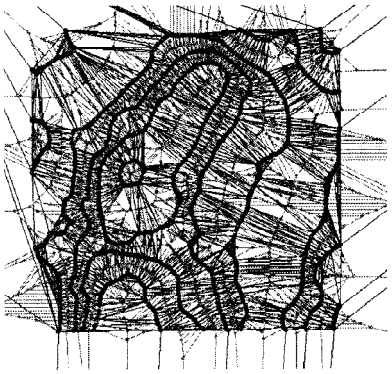


Fig. 23. A contour map.

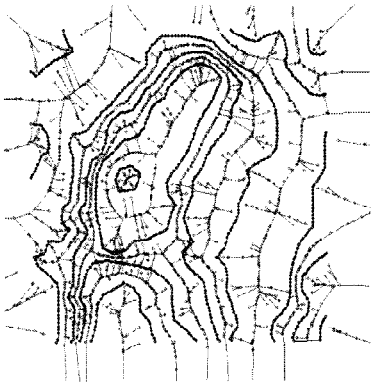


Fig. 24. Crust and skeleton.

derived contours from stereo images are readily available. The combination of the crust (contours) and the skeleton assists this process considerably, as the skeleton expresses the relationships between contours or between contour segments. Figures 23 and 24 show this well. In general, the skeleton forms a medial contour, but where contour re-entrants occur, branches of the skeleton indicate that the associated dual Delaunay edge would have connected vertices of the same elevation—thus giving an implausible horizontal triangular plate. Various solutions are possible, including the treatment of these branches as gullies, having elevations varying between the medial axis and the associated contour line. This follows closely the interpretation that would be made by the experienced map user. Cleanup of the "hair" could first be performed as described previously. This gives a very powerful tool for interpreting the terrain on the basis of the relationships between contours, and not merely by individual contours themselves. The Voronoi/Delaunay relationships are particularly useful here, as there is zero slope along contour lines defined by Delaunay (crust) edges, and thus the direction of runoff or maximum downhill slope is perpendicular to this—exactly as expressed by the associated dual Voronoi edge. These inter-relationships readily permit the reconstruction of a meaningful terrain model, as they mimic the natural processes involved.

7. Conclusions and Future Work. We have shown that a simple one-step algorithm may generate the crust and anti-crust simultaneously, and that these may be extracted to form topologically structured maps. The results are equivalent to those of the crust algorithm of [ABE]. We have shown how this resolves a variety of issues in map input and analysis, and we expect to address individual applications in more detail in the near future.

In two dimensions the crust is a set of line segments, as is the skeleton. Amenta et al. [AB], [ABK] describe the case for three-dimensional crust extraction, which is similar to that in two dimensions. However, they note that certain Voronoi vertices do not lie on the skeleton, and therefore need to be excluded in their algorithm. Our one-step algorithm also applies here: the choice for a simplex-pair is either a triangular face assigned to the crust, or a line segment assigned to the skeleton. We have not yet completed our exploration of this modification of the crust approach.

The Quad-Arc approach may be of significant use in Geomatics as it provides a simple data structure that suits the needs of the data input process, preserves the map topology, and may be implemented within small-scale PC-based software for topological map querying. The combination of the crust criterion, the one-step algorithm, and the Quad-Arc data structure are sufficient to put cartographic topology within the range of simple mapping systems.

Acknowledgments. We thank Nina Amenta for her comments and encouragement, and for pointing out the link to Attali's work. David Thibault prepared many of the diagrams and assisted with the programming. Mir Mostafavi, Darka Mioc, Francois Anton and Weiping Yang assisted at various stages of the development of the underlying programs.

References

[A] Attali, D. (1997). r-Regular shape reconstruction from unorganized points. *Proc*. 13*th Annual ACM Symposium on Computational Geometry*, pp. 248–253.

[AB] Amenta, N., and M. Bern (1998). Surface reconstruction by Voronoi filtering. *Proc*. 14*th Annual ACM Symposium on Computational Geometry*, pp. 39–48.

[ABE] Amenta, N., M. Bern and D. Eppstein (1998). The crust and the beta-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, **60/62**(2), 125–135.

[ABK] Amenta, N., M. Bern and M. Kamvysselis (1998). A new Voronoi-based surface reconstruction algorithm. *Proc. Siggraph* '98, pp. 415–421.

[AS] Alt, H., and O. Schwarzkopf (1995). The Voronoi diagram of curved objects. *Proc*. 11*th Annual ACM Symposium on Computational Geometry*, pp. 89–97.

[B1] Blum, H. (1967). A transformation for extracting new descriptors of shape. In: W. Whaten-Dunn (ed.), *Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, Mass., pp. 153–171.

[B2] Burrough, P. A. (1986). *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford University Press, New York.

[BA] Brandt, J., and V. R. Algazi (1992). Continuous skeleton computation by Voronoi diagram. *Computer Vision, Graphics and Image Processing*, **55**, 329–338.

[BKO] de Berg, M., M. van Kreveld, R. van Oostrum and M. Overmars (1997). Simple traversal of a subdivision without extra storage. *International Journal on Geographical Information Systems*, **11**, 359–373.

[BM1] Burge, M., and G. Monagan (1995). Using the Voronoi tessellation for grouping words and multi-part symbols in documents. *Proc. Vision Geometry* IV, SPIE, vol. 2573 (San Diego, California), pp. 116–123.

[BM2] Burge, M., and G. Monagan (1995). Extracting words and multi-part symbols in graphics rich documents. *Proc*. 8*th International Conference on Image Analysis and Processing* (San Remo). LNCS, Springer-Verlag, Berlin, pp. 533–538.

[CDW] Chrisman, N. R., J. R. Dougenik and D. White (1992). Lessons for the design of polygon overlay processing from the Odyssey Whirlpool algorithm. *Proc*. 5*th International Symposium on Spatial Data Handling*, pp. 401–409.

[F] Frank, A. (1987). Overlay processing in spatial information systems. *Proc*. *Auto-Carto* 8, pp. 16–31.

[FFNP] De Floriani, L., B. Falcidieno, G. Nagy and C. Pienovi (1991). On sorting triangles in a Delaunay tessellation. *Algorithmica*, **6**, 522–532.

[G1] Gold, C. M. (1992). The meaning of "neighbour." In: *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space* LNCS 639. Springer-Verlag, Berlin, pp. 220–235.

[G2] Gold, C. M. (1994). Dynamic data structures: the interactive map. In: M. Molenaar and S. de Hoop (eds.), *Advanced Geographic Data Modelling—Spatial Data Modelling and Query Languages for* 2*D and* 3*D Applications*, Netherlands Geodetic Commission Publications on Geodesy (New Series), pp. 121–128.

[G3] Gold, C. M. (1994). Three approaches to automated topology, and how computational geometry helps. In: T. C. Waugh and R. G. Healey (eds.), *Proc. Sixth International Symposium on Spatial Data Handling* (Edinburgh, 1994), pp. 145–158.

[G4] Gold, C. M. (1997). Simple topology generation from scanned maps. *Proc. Auto-Carto* 13, pp. 337–346.

[G5] Gold, C. M. (1998). The Quad-Arc data structure. In: T. K. Poiker and N. R. Chrisman (eds.), *Proc. 8th International Symposium on Spatial Data Handling* (Vancouver, BC, 1998), pp. 713–724.

[GCR] Gold, C. M., T. D. Charters and J. Ramsden (1977). Automated contour mapping using triangular element data structures and an interpolant over each triangular domain. *Computer Graphics*, **5**, 170–175.

[GC] Gold, C. M., and S. Cormack (1987). Spatially ordered networks and topographic reconstructions. *International Journal of Geographical Information Systems*, **1**, 137–148.

[GNY] Gold, C. M., J. Nantel and W. Yang (1996). Outside-in: an alternative approach to forest map digitizing. *International Journal of Geographical Information Systems*, **10**(3), 291–310.

[GRR] Gold, C. M., P. R. Remmele and T. Roos (1997). Voronoi methods in GIS. In: M. van Kreveld, J. Nievergeld, T. Roos and P. Widmeyer (eds.), *Algorithmic Foundations of GIS*. LNCS 1340. Springer-Verlag, Berlin, pp. 21–35.

[GS] Guibas, L., and J. Stolfi (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *Transactions on Graphics*, **4**, 74–123.

[GSS] Guibas, L., D. Salesin and J. Stolfi (1989). Epsilon geometry: building robust algorithms from imprecise computations. *Proc.* 5*th Annual ACM Symposium on Computational Geometry*, pp. 208–217.

[H] Herring, J. (1989). A fully integrated geographical information system. *Proc. Auto-Carto* 9, pp. 828–837.

[NME] Nagy, G., M. Mukherjee and D. Embly (1990). Making do with finite numerical precision in spatial data structures. *Proc.* 4*th International Symposium on Spatial Data Handling*, pp. 55–65.

[O] Ogniewicz, R. L. (1994). Skeleton-space: a multiscale shape description combining region and boundary information. *Proc. Computer Vision and Pattern Recognition*, pp. 746–751.

[OBS] Okabe, A., B. Boots and K. Sugihara (1992). *Spatial Tessellations—Concepts and Applications of Voronoi Diagrams*. Wiley, Chichester.

[OI] Ogniewicz, R., and M. Ilg (1990). Skeletons with Euclidean metric and correct topology and their application in object recognition and document analysis. *Proc.* 4*th International Symposium on Spatial Data Handling*, vol. 1, pp. 15–24.

[P] Pullar, D. (1994). A tractable approach to map overlay. Ph.D. Thesis, University of Maine, Orono, Maine.

[PC] Peucker, T. K., and N. R. Chrisman (1975). Cartographic data structures. *American Cartographer*, **2**, 55–69.

[RCGH] Robinson, G. P., A. C. F. Colchester, L. D. Griffin and D. J. Hawkes (1992). Integrated skeleton and boundary shape representation for medical image interpretation. *Proc. European Conference on Computer Vision*, pp. 725–729.

[S] Sedgewick, R. (1983). *Algorithms*. Addison-Wesley, Reading, Mass.

[SS] Speckmann, B., and J. S. Snoeyink (1997). Easy triangle strips for TIN terrain models. *Proc.* 9*th Canadian Conference on Computational Geometry*, pp. 239–244.

[W] Worboys, M. F. (1995). *GIS: A Computing Perspective*. Taylor and Francis, London.