Computing the Medial Axis Transform of Polygonal Domains by Tracing Paths

R. Joan-Arinyo, L. Pérez, J. Vilaplana
Dept. Llenguatges i Sistemes Informàtics.
Universitat Politècnica de Catalunya

e-mail [robert, lpv, josep]@lsi.upc.es

Abstract

A practical algorithm for computing the medial axis transform of 2D polygonal domains is presented. The algorithm computes the medial axis by tracing its paths. The medial axis is generated in the form of a graph where nodes are medial axis keypoints and edges are medial axis paths bounded by keypoints. Graph edges are labeled with the boundary elements governing the local medial axis shape. Examples with different domain boundaries illustrate the algorithm performance.

1 Introduction

The Medial Axis Transform (MAT) was introduced by Blum, [3], as a means to describe shapes in biology and medicine. Currently, the MAT has become an important descriptive tool in several fields like modelling growth, path planning, feature recognition, and finite element mesh generation. Since it provides a complete representation of a solid, it can be used as an alternative representation in solid modellers.

Several algorithms to determine the medial axis from a 2D polygonal boundary have been proposed. Some algorithms [6, 7, 11, 12] use an offset process of the boundary directed towards the interior of the object to compute the medial axis. This process is analogous to the propagation at uniform speed of a grass-fire wavefront towards the interior of a region. The locus of points where the wavefront meets itself corresponds to the medial axis of the boundary region. Based on this idea, Montanari [11] developed an algorithm for connected polygonal figures. Preparata [12] presented an $O(n\ log\ n)$ algorithm for computing the medial axis of a convex polygon and an $O(n^2)$ algorithm for concave polygons. Gursoy and Patrikalakis [6, 7] developed an algorithm to compute the medial axis of multiply connected planar regions bounded by line segments, circular arcs and general nonuniform rational B-splines. They also proposed an automatic finite element mesh generation from the medial axis. Some other algorithms are based on Voronoi diagram determination. Lee [10] developed an $O(n\ log\ n)$ divide and conquer algorithm for simple polygons. It begins with the computation of the Voronoi diagram for the polygonal domain and then removes Voronoi edges at the concave vertices of the domain to obtain the medial axis. Another related idea is the derivation of the medial axis from the Delaunay triangulation, [8, 14]

Other works use discrete and approximate techniques for computing the medial axis. Some of them can be found in [4, 5, 8, 9] These works differ in the boundary representation used. Continuous medial axis for 3D polyhedra can be found in [15, 16]. A recent and thorough review of work on MAT can be found in [16].

The efforts that the scientific community has devoted to the medial axis have resulted in a large number of papers in the literature. However, the effective use of the medial axis has been hampered by the lack of practical and robust algorithms [2].

The present work reports on a simple and reliable algorithm that generates the medial axis of polygonal domains. The algorithm is based on the fact that the locus of centers of all maximal discs inscribed in the polygonal boundary defines the medial axis. The medial axis axis is generated by tracing paths in it, that is, by computing connected subsets of points of the medial axis. The shape of the subset is defined by a subset of boundary elements. The medial axis is generated in the form of a graph where nodes are medial axis keypoints and edges are medial axis paths bounded by keypoints. Graph edges are labeled with the boundary elements governing the medial axis path shape. The resulting algorithm is conceptually simple and easy to extend to 3D polyhedra.

The paper is organized as follows. Section 2 provides the basic definitions. Section 3 describes in detail the algorithm. Section 4 presents some examples which illustrate the behaviour of the algorithm. Finally, Section 5 summarizes the paper and suggests further work lines.

2 Basic Definitions

In this Section we give some definitions and describe the properties of the medial axis relevant to the algorithm presented in Section 3.

2.1 Medial Axis Transform

Here we give several definitions that are standard in the field, [3, 17].

Definition 2.1 *A disc is said to be maximal in a planar object if it is contained in the object but is not a proper subset of any other disc contained in the object.*

Definition 2.2 *The medial axis of a planar object is the locus of the centers of all maximal discs in the object, with the limit points of this locus.*

Definition 2.3 *The radius function of the medial axis of a domain is a continuous, real-valued function defined on the medial axis whose value at each point is equal to the radius of the maximal disc centered in the point.*

Definition 2.4 *The Medial Axis Transform of a planar object is its medial axis along with its associated radius function.*

In this paper, we consider polygonal domains in a 2-D Euclidean space with arbritrary genus, straight edges with no two adjacent, colinear edges in the boundary. In this context, the maximal

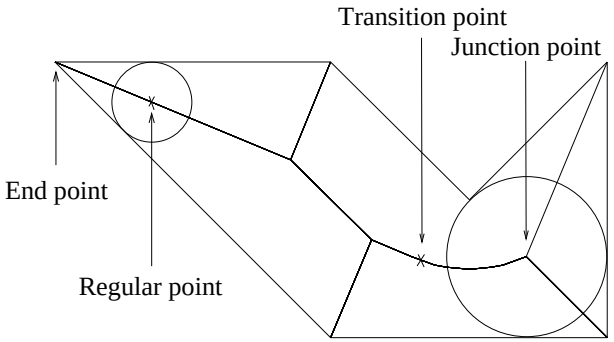

Figure 1: Taxonomy of medial axis points.

discs that define the medial axis can be tangent only to edges and concave vertices in the boundary. We will refer to the edges and concave vertices in the domain boundary as the *active boundary elements*. The subset of active active boundary elements to which a maximal inscribed disc centered on a medial axis point is tangent to are called the medial axis point *governors*, [16].

Points in the medial axis can be classified according to the number of governors that define them. The classification is as follows (see Figure 1):

1. *Junction points:* The maximal disc is tangent to three or more governors.
2. *End Points:* An end point results from the intersection of the medial axis with the domain boundary. These points are actually limit points of the medial axis where the radius function has a value of zero. In our context, these points are the convex vertices of the polygon.
3. *Regular Points:* Are those points where the maximal disc is tangent to two different governors. Points in the medial axis that are neither junction points nor end points are regular points.

Additionally, we define a *transition point* as a medial axis regular point where one of both governors changes. We refer generically to end points, junction points and transition points as *key points*.

We define a *path* as the subset of the medial axis such that all its points are generated by the same set of governors. A path is bounded by two keypoints, one at each end.

2.2 Influence Regions and Path Shapes

A point in the medial axis is equidistant from two or more active elements on the domain boundary. This fact results in the existence of strong relations between the medial axis and sets of points with some equidistance function applied on them. Among all these point sets, the Voronoi diagrams are particularly useful for our purposes.

Consider a collection of sets s_i defined in 2-D Euclidean space. The *Voronoi region*, $VR(s_i)$, associated with s_i, is the locus of points closer to s_i than to any other set. The *Voronoi diagram* of the collection, denoted as $VD(s_i)$, is the locus of points in the space belonging to two or more Voronoi regions.

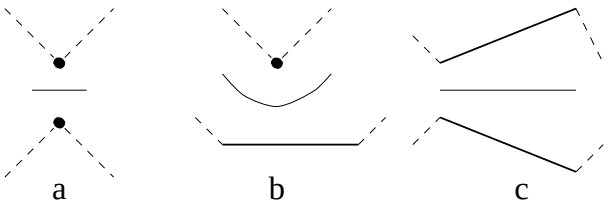

Figure 2: Generalized Voronoi diagrams for polygon elements.

When the set s_i consists of one point, the associated Voronoi region is a polygon known as the *Voronoi polygon*, and the Voronoi diagram consists of vertices, named *Voronoi vertices*, and line segments known as *Voronoi edges*, [2, 13]

Since we focus on polygonal objects, we will consider that sets s_i are made from points and open, straight segments. They will correspond respectively to concave vertices and open, straight edges found in the polygon boundary. In this context, the generalized Voronoi diagrams associated with every pair of governors are (see Figure 2):

1. *Concave vertex-Concave vertex* pair: straight bisector of the segment defined by the pair of points. Figure 2a.
2. *Edge-Concave vertex* pair: arc of the parabola defined by the concave vertex as its focus and the edge as its directrix. Figure 2b.
3. *Edge-Edge* pair: The straight bisector of the convex angle formed by the intersection of the supporting lines of the edge pair. Figure 2c.

The concept of *influence region* of a governor defined in [8] plays an important role in our algorithm. We recall the definition, illustrated in Figure 3, here just for completeness.

Definition 2.5 *Let e be an edge of a polygonal domain boundary. Let $V(e)$ be its Voronoi polygon and let (ve_i, ve_j) be the pair of edges in $V(e)$ through the endpoints of e. The influence region of e is the set of points bounded by edge e and the two half-lines supporting the Voronoi edges (ve_i, ve_j).*

Definition 2.6 *Let v be a concave vertex on a polygonal domain boundary. Let $V(v)$ be its Voronoi polygon and let (vv_i, vv_j) be the pair of edges in $V(v)$ incident to the vertex v. The influence region of v is the set of points bounded of v and the two halflines supporting the Voronoi edges (vv_i, vv_j).*

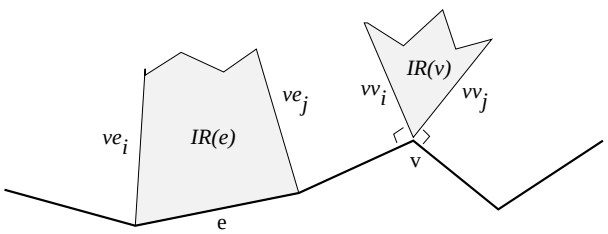

Figure 3: Influence regions of straight segments and concave vertices in a 2D polygonal domain boundary.

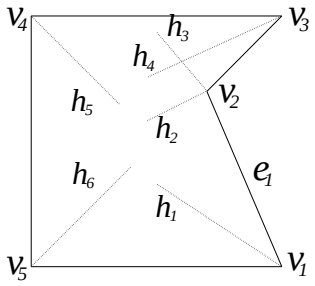

Figure 4: Oriented boundary and induced order in the set of Voronoi halflines.

From these definitons, it is obvious that a given governor can only determine the shape of the medial axis in its own influence region. As we shall see in the next Section, this fact is exploited by our algorithm.

2.3 Orientation

We assume that the polygonal domain external boundary is oriented, anticlockwise, for example. According to this orientation, for each edge e_i in the active boundary elements set, we distinguish an initial and a final vertex, $e_i(v_1)$ and $e_i(v_2)$. Let us denote by h_1 and h_2 the halflines which support the Voronoi edges associated with the initial and final vertex respectively of a given edge in the domain boundary. The orientation of the domain boundary naturally induces an order in the Voronoi halflines set. Hence the pair of Voronoi halflines associated with each concave vertex considered as an active boundary element is also an ordered set. See Figure 4. Therefore we will consider concave vertices as being oriented and we will denote by $v_i(h_1)$ and $v_i(h_2)$ the ordered Voronoi halflines associated with a concave vertex v_i. In general, we will refer to h_1 as the *back* halfline and to h_2 as the *front* halfline.

3 The Algorithm

The medial axis is generated in the form of an undirected graph, $G(V, E)$, where each vertex in the set V is a keypoint, and an edge in E is a medial axis path connecting two keypoints. Edges are labeled with the governors of the corresponding path. Figure 5 shows an example.

Assume that paths in the medial axis are traced by an inscribed disc moving from the initial keypoint towards the final keypoint while its radius continuously changes in such a way that the disc is always tangent to the governors. It is easy to see that there is one governor which has its orientation coincident with the disc center motion while the orientation of the other governor is opposite. See Figure 6. We will refer to this property by saying that a governor is either *isooriented* or *counteroriented*.

The algorithm computes the medial axis by tracing each component path. Assume that the initial keypoint of a path and its governors are known. Since the set of governors defines the path shape, the path is traced just by computing the final keypoint in the path. Let `d` be a disc centered on a medial axis point and tangent to the properly oriented governors `g1` and `g2`. The algorithm can be stated as follows.

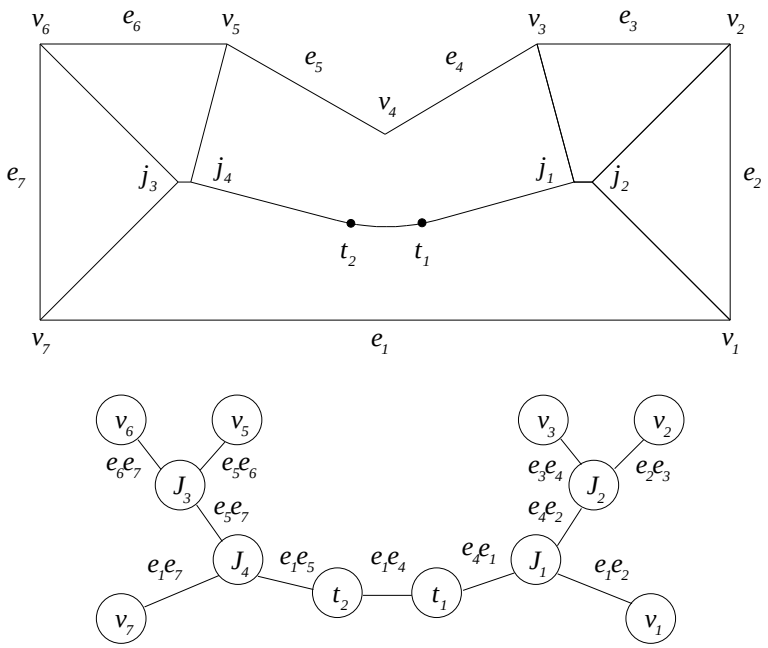

Figure 5: Medial axis of a polygonal domain and associated graph.

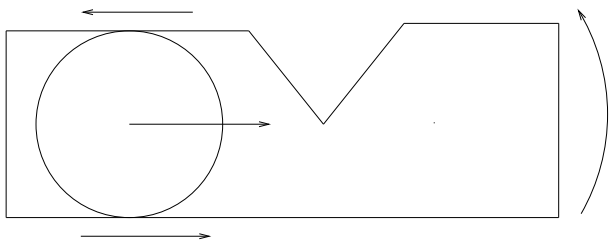

Figure 6: Governors orientation with respect to the disc motion.

```
algorithm MAbyTracingPaths
  InitializeFirstPath (d, g1, g2)
  PushPath (d, g1, g2)
  while not EmptyPath() do
    PopPath (d, g1, g2)
    TracePath (d, g1, g2, InterferenceList)
    if not EndDisc(d) then
        InitializeNewPaths (d, g1, g2, InterferenceList, NewPathList)
        for path in NewPathList do
          PushPath (d, g1, g2)
        endfor
    endif
  endwhile
endalgorithm
```

We develop first the procedure to trace a medial axis path. Then we will show how to initialize the first path and the subsequent new paths.

3.1 Tracing a Medial Axis Path

Assume that a maximal inscribed disc centered at the initial keypoint of the path to be traced and the corresponding governors are known. To trace the path, all we need is to compute the final keypoint in the path. This is performed in two steps. First a final keypoint candidate is computed. Then we check for its validity. If the candidate is valid we are done; otherwise, from the information of the validity check, a new keypoint candidate is computed, which is more likely to be a valid keypoint. Let us show how computations are carried out depending on the type of the governors of the path to be traced.

Edge-Edge case

As pointed out in Section 2.2, when the path to be traced is governed by two straight edges the medial axis is the straight segment that bisects the angle spanned by the straight lines supporting the governors.

Let e_i be the counteroriented governor and e_j be the isooriented governor. Let h_{i1} the back Voronoi halfline associated with governor e_i and h_{j2} the front Voronoi halfline associated with governor e_j. Furthermore, let l_m be the straight line that supports locally the medial axis. See Figure 7.

Since the medial axis points must belong to the intersection of the influence regions of its governors, the path cannot extend beyond any one of the points m_i or m_j where h_{i1} and h_{j2} respectively intersect l_m. The candidate keypoint is the intersection point closest to the initial keypoint in the path. The disc radius is given by the perpendicular distance from the path final keypoint to any of the governors.

After a keypoint candidate is computed, we check whether it is valid or not. We study whether there is any active boundary element which intersects the maximal disc centered at the candidate keypoint. If there are no intersections, the keypoint is valid; otherwise the intersecting boundary element closest to the path starting keypoint together with the path governors define a new keypoint

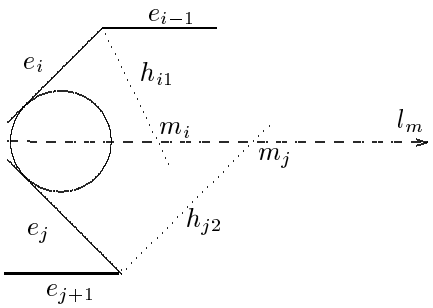

Figure 7: Edge-Edge case.

candidate with its maximal inscribed disc, [1]. In many cases, the element closest to the path will become a new governor, i. e. the keypoint candidate will be a valid keypoint. Otherwise, the procedure is repeated with the new intersecting boundary.

Concave Vertex-Edge case

This case is illustrated in Figure 8. Now, the pair of governors are a boundary edge, e_j, and a concave boundary vertex, v_i. Therefore the path defined in the medial axis is an arc of parabola.

The path final keypoint candidate is computed in a similar way as the edge-edge case. Note that the Voronoi halflines associated with vertex v_i have their origin in this same vertex, therefore each of them intersects the parabolic path in just one point.

Concave vertex-Concave vertex case.

When the governors are two concave vertices, the medial axis path is a straight segment perpendicular to the straight line joining the two concave vertices through its midpoint. Let v_i and v_j be the governors. Without loss of generality, assume that v_i is isooriented and v_j is counteroriented. Moreover, assume that h_{i1} and h_{j2} are the front and back Voronoi halflines associated respectively with v_i and v_j. Here we distinguish two different situations as shown in Figure 9.

If at least one of the halflines h_{i1} or h_{j2} intersects the path, the path final keypoint is determined in the same way as in the previous cases.

If neither h_{i1} nor h_{j2} intersect the path, we need another active boundary element acting as a third governor candidate. This new governor candidate is the active boundary element closest to the last computed point in the medial axis. If there are several active boundary elements at the same minimum distance, all of them are taken into account because all together could define a medial axis junction point. From here, the remaining procedure is the same as the other cases.

3.2 Initialization of the First Path

To initialize the first path, a starting keypoint in the medial axis and the corresponding governors must be determined. From the definitions given in Section 2, we know that all convex vertices in the domain boundary are end points in the domain medial axis. Therefore, we choose any convex vertex as the starting point and the edges incident on it, properly oriented, as the governors.

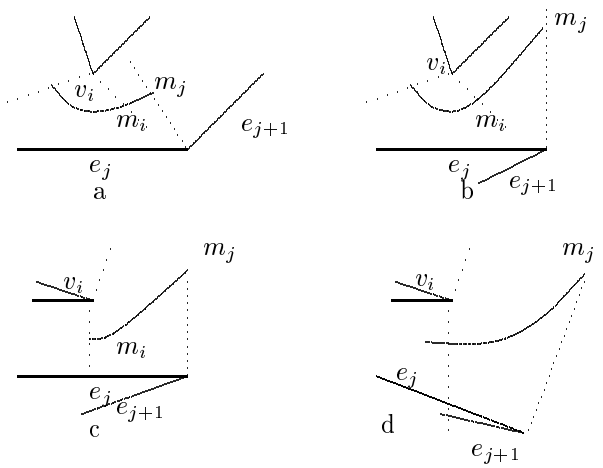

Figure 8: Concave Vertex-Edge case.

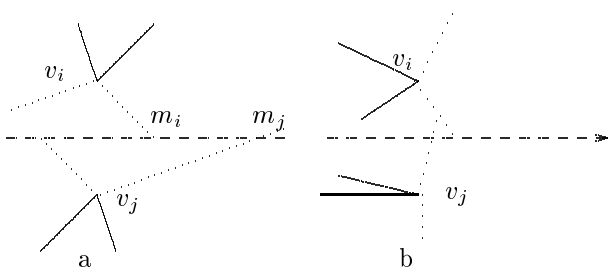

Figure 9: Concave vertex-Concave vertex case.

3.3 Initialization of New Paths

Once the initial path has been traced, new paths start at junction points and at transition points.

A maximal disc centered on a junction point is tangent to three or more active boundary elements. Therefore, junction points are starting points of two or more new paths. The set of active boundary elements defining a junction point is sorted circularly with respect to the junction point. The first and last elements in the set are the governors which defined the path leading to the junction point. The junction point along with each pair of consecutive active boundary elements define a new path which is pushed into the paths stack.

Every time a medial axis path reaches a transition point, the set of governors must be updated. In all cases, there is at least one concave vertex involved, either because a concave vertex in the active boundary elements set becomes a governor or, viceversa, a concave vertex is no longer one of the governors. Determining the new governors is straightforward because they are adjacent to the old ones in the domain boundary.

4 Examples

We illustrate the behaviour of the algorithm giving some examples.

In the first example, depicted in Figure 10, the algorithm computes the medial axis of a convex poligonal domain. The initial path is defined by convex vertex v_1 and the two edges incident on it as governors, marked with bold lines. The bisector of the governors is the path for the current disc. The final keypoint is computed as explained in the edge-edge case in Section 3. See Figure 10b. When checking for candidate keypoint validity, the algorithm finds that the maximal disc is tangent to three governors defining a junction point. At this point, two new paths are initialised: One is defined by edges $\overline{v_1v_2}$ and $\overline{v_4v_5}$; the other by edges $\overline{v_5v_1}$ and $\overline{v_4v_5}$. Both paths are pushed into the stack. Then the algorithm pops the path defined by governors $\overline{v_1v_5}$ and $\overline{v_4v_5}$ and traces it. This is performed computing the final path keypoint as the intersection of the governors, i.e., vertex v_5 in Figure 10c. Figure 10d and Figure 10e show maximal discs centered on junction points. Finally, Figure 10f gives the complete medial axis.

An example of a concave polygon is given in Figure 11. Assume that the algorithm has traced part of the medial axis as shown in Figure 11a. The final keypoint candidate is computed as stated in the edge-concave vertex case in Section 3. In this case, the disc centered on the keypoint candidate is seen to intersect several boundary elements, for example edge $\overline{v_qv_r}$, resulting in an invalid candidate. This test gives a list of active boundary elements which intersect the disc. The new path final keypoint is defined by the old governors and the element of that list closest to the initial keypoint of this path, as shown in Figure 11c. The center of the disc in Figure 11d is a junction point from where the medial axis computation proceeds,

The last example, given in Figure 12, shows concave vertices whose Voronoi edges cannot bound the computed paths. In Figure 12a the medial axis reaches a transition point whose disc is tangent to the two bottom concave vertices that will be the new governors. The search for minimum distance element(s) to the transition point yields both top concave vertices. The four related points define a new junction point of the medial axis (Figure 12b) where three new paths start. Figure 12c shows how the algorithm progresses.

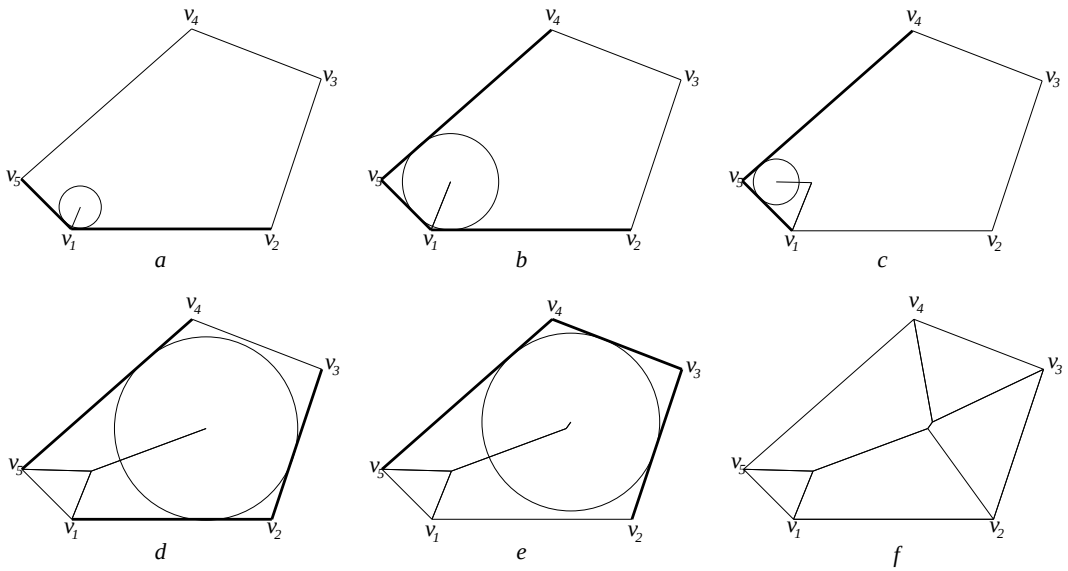

Figure 10: MAT computation for an example of a convex polygon.

5 Summary

An algorithm for computing the medial axis transform of 2D polygonal domains has been presented. The algorithm computes the medial axis by tracing paths bounded by points where the set of governors change. The medial axis is generated as a graph whose nodes are the medial axis points where the set of governors change and whose edges are labeled with the set of governors which define each path shape. It is conceptually clear, and easy to implement.

Future work will focus on the design of a suitable data structure to improve the efficiency of the algorithm presented here. A fast detection of active boundary elements that intersect the influence regions of the set of governors under study will improve the algorithm performance.

The concepts used in 2D space can be generalized to design an algorithm for deriving the medial axis transform of 3D polyhedra. In 3D space, the maximal discs will be spheres and the governors will be faces, edges and vertices of the polyhedron. Path shapes will be planar, parabolic, and hyperbolic surfaces. The limits of these shapes will be traced by the center of maximal spheres moving between keypoints.

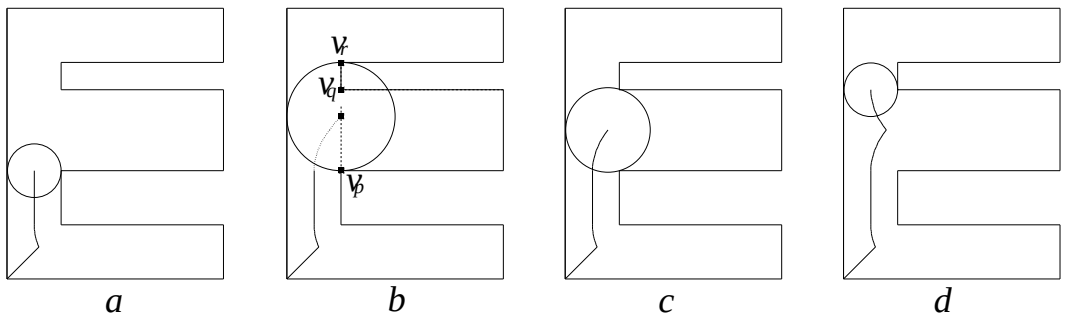

Figure 11: MAT computation for an example of a concave polygon

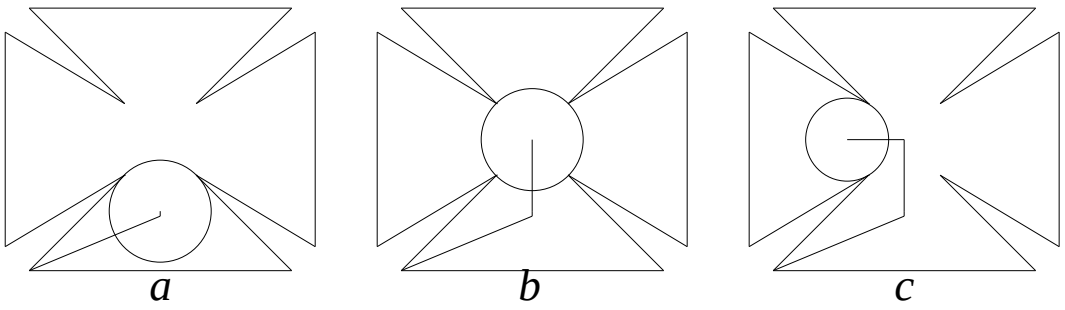

Figure 12: Concave vertices that do not bound a path.

Acknowledgements

This work has been partially supported by the CICYT, The Spanish Research Agency, under grants TIC95-0630-C05-04 and TIC95-0630-C05-05.

References

[1] C. Armstrong, T. Tam, D. Robinson, R. McKeag, and M. Price. Automatic generation of finite element meshes. In *SERC ACME Directorate Research Conference*, England, 1990.

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

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

[4] J. W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CGVIP: Image Understanding*, 59(1):116–124, January 1994.

[5] P.-E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[6] H. Nebi Gürsoy and Nicholas M. Patrilakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part I algorithms. *Engineering with Computers*, 8(3):121–137, 1992.

[7] H. Nebi Gürsoy and Nicholas M. Patrilakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part II implementation. *Engineering with Computers*, 8(4):179–196, 1992.

[8] R. Joan-Arinyo, L. Pérez-Vidal, and E. Gargallo-Monllau. An adaptive algorithm to compute the medial axis transform of 2D polygonal domains. In P. Brunet and D. Roller, editors, *CAD Tools for Products*. Springer Verlag, 1996.

[9] D. Lavender, A. Bowyer, J. Davenport, A. Wallis, and J. Woodwark. Voronoi diagrams of set-theoretic solid models. *IEEE Computer Graphics and Applications*, 12(5):69–77, September 1992.

[10] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. and Mach. Intelligence*, PAMI-4:363–369, 1982.

[11] U. Montanari. Continuous skeletons from digitized images. *JACM*, 16:534–549, 1969.

[12] F. Preparata. The medial axis of a simple polygon. In *Proc. 6th Symp. Mathematical Foundations of Comp. Sci.*, pages 443–450, 1977.

[13] F. Preparata and M. Shamos. *Computational Geometry.* Springer Verlag, New York, 1985.

[14] M. Price, T. Tam, C. Armstrong, and R. McKeag. Computing the branch points of the Voronoi diagram of a object using a point Delaunay triangulation algorithm. Draft manuscript, 1991.

[15] Jayachandra M. Reddy and George M. Turkiyyah. Computation of 3D skeletons using a generalized Delaunay triangulation technique. *Computer-Aided Design*, 27(9):677–694, 1995.

[16] E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3-D polyhedra. In C.M. Hoffmann and J. Rossignac, editors, *Third Symposium on Solid Modeling and Applications*, pages 187–199, Salt Lake City, Utah, 17-19 May 1995. ACM Press.

[17] F.-E. Wolter. Cut locus and medial axis transform in global shape interrogation and representation. Memorandum 92-2, MIT Ocean Engineering Design Laboratory, January 1992.