ELSEVIER

Computer-Aided Design 35 (2002) 619–632

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Constructing medial axis transform of planar domains with curved boundaries

M. Ramanathan, B. Gurumoorthy*

Department of Mechanical Engineering, Indian Institute of Science, Bangalore 560 012, India

Received 11 July 2001; revised 13 March 2002; accepted 20 March 2002

Abstract

The paper describes an algorithm for generating an approximation of the medial axis transform (MAT) for planar objects with free form boundaries. The algorithm generates the MAT by a tracing technique that marches along the object boundary rather than the bisectors of the boundary entities. The level of approximation is controlled by the choice of the step size in the tracing procedure. Criteria based on distance and local curvature of boundary entities are used to identify the junction or branch points and the search for these branch points is more efficient than while tracing the bisectors. The algorithm works for multiply connected objects as well. Results of implementation are provided.

Keywords: Medial axis transform; Voronoi diagram; Free form boundaries

1. Introduction: medial axis transform

The medial axis transform (MAT) was first introduced by Blum [1,2] to describe biological shape. It can be viewed as the locus of the center of a maximal disc as it rolls inside an object. Since its introduction, the MAT has found use in a wide variety of applications that primarily involve reasoning about geometry or shape. The MAT has been used in pattern analysis and image analysis [3,4], finite element mesh generation [5,6], mold design [7] and path planning [8] to name a few. There is a one-to-one correspondence between the MAT and object boundary which means that for an object there will be a unique MAT. Moreover, it is possible to reconstruct the object given its MAT [9]. The MAT can potentially be used as a representation scheme in geometric modelers, along with more popular schemes, such as constructive solid geometry (CSG) and boundary representation (B-rep) [10]. More importantly, dimensional reduction and topological equivalence make the MAT a simplified, abstract representation of the geometry. Since the MAT also provides details with respect to the symmetry of the object, it can be applied to application areas where this property is considered important.

The medial axis (MA), or skeleton of the set D, denoted M(D), is defined as the locus of points inside D which lie at the centers of all closed discs (or balls in 3-D) which are maximal in D, together with the limit points of this locus. A closed disc (or ball) is said to be *maximal* in a subset D of the 2-D (or 3-D) space if it is contained in D but is not a proper subset of any other disc (or ball) contained in D. The radius function of the MA of D is a continuous, real-valued function defined on M(D) whose value at each point on the MA is equal to the radius of the associated maximal disc or ball. The MAT of D is the MA together with its associated radius function.

The boundary and the corresponding MAT of an object is shown in Fig. 1. If the boundary segments of the object consists of only points, straight line segments and circular arcs, then the MAT segments will be one of the conic sections [11]. An important characteristic of the MAT is that it can be used to simplify the original object and still retain the original object's information. For instance, the 2-D MAT defines a unique, coordinate-system-independent decomposition of a planar shape into lines and the 3-D MAT simplifies a solid model into a collection of surface patches. MAT of an object is therefore also called the *skeleton* or *symmetric axis transform* of a part/shape.

Properties of MAT include:

- *Uniqueness* [24]. There is an unique MAT for a given object.

* Corresponding author.
E-mail addresses: bgm@mecheng.iisc.ernet.in (B. Gurumoorthy); raman@mecheng.iisc.ernet.in (M. Ramanathan).

0010-4485/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(02)00085-4

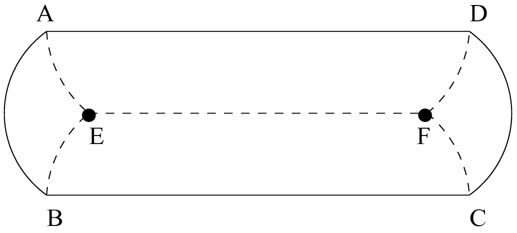
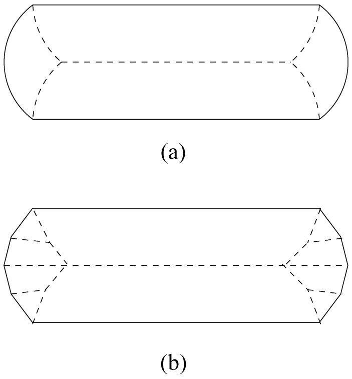


Fig. 1. Boundary and its medial axis.

- *Invertibility* [9]. With the axis and its radius function one can reconstruct the object by taking the union of all circles centered on the points corresponding to the axis, each with a radius given by radius function.
- *Dimensional reduction* [6]. The dimensionality of a MAT is lower than that of its object.
- *Topological equivalence* [24]. A MAT is topologically equivalent to its object.
- *Symmetry information* [26]. MAT establishes an object centered representation from which individual shape characteristics can be measured.
- *One-to-one correspondence* [22]. There exists an one-to-one correspondence between the object boundary and its MAT.

There have been several efforts reported for the construction of MAT. These approaches can be classified based on the type of representation of the part used in the construction and on the nature of the description of MAT constructed. A continuous description of the MAT is possible from a continuous representation of the part only for the case of polygons where the boundary consists of linear segments. For a part with nonlinear or curved segments a continuous description of MAT is only possible if the part representation is discretized. This, however, results in artificial MAT segments that are artifacts of the representation used and not due to the underlying part geometry (see Fig. 2) [12]. For objects with free-form entities, a continuous description of the MAT is possible only if it is possible to obtain a continuous description of the bisector of a pair of free-form curved edges. For a pair of edges in a plane the bisector is a set of points which are equidistant from the two edges. It is possible to obtain a continuous description of the bisector only for a few special cases—a point and a rational curve in a plane for which case the bisector is a rational curve [13] and two rational space curves for which the bisector is a rational surface [14]. However, for the case of coplanar curves (polynomial or rational), the bisector has been shown to be algebraic but not rational [13]. This has resulted in the need for numerical tracing of the bisector curves (or surfaces) [15,16] which is computationally expensive [14]. In addition to this, the topology of the final MAT has to be ensured by correctly trimming each bisector segment and connecting the valid segments. This adds to the substantial effort already required in generating the bisector segments. For example, the dashed line in Fig. 3 (where B(a,b) denotes the bisector between entities a and b on the boundary of the domain) shows the bisector curve for the pair of edges AB and CD. The bisector consists of several segments some of which may be nonlinear (segment P3-P4 in the figure corresponds to the set of points equidistant from point B in edge AB and the edge CD). For the same pair of line segments AB and CD, when it is part of the object ABCD, B1–B2 is the corresponding MAT segment, requiring trimming of the rest of the segments forming the bisector.


(a)

(b)

Fig. 2. Effect of discretizing boundary on MAT.

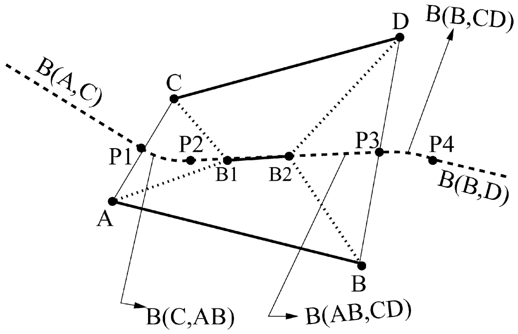


Fig. 3. Difference between MAT and bisector.

The choice of the part representation used and the type of description of MAT obtained depends on the application that the MAT is to be used for. For applications such as finite element mesh generation and path planning a discretized representation of the object could be used to obtain a continuous MAT. It must be noted that even here some effort is required in trimming and post-processing the continuous MAT to be in conformity with the topology of the free-form object. The additional MAT segments will distort the reasoning based on MAT (see Fig. 2, where the MAT is substantially different when the exact representation is used). Exact representation of the part has to be used in applications where the geometry of the part has to be reconstructed from the MAT [9]. Using the exact representation of the part for constructing the MAT also eliminates the need for additional processing required to eliminate the artificial segments that may arise due to the discretization [12].

In this paper, the problem of generating a MAT for planar objects with free-form curved entities has been addressed. The algorithm proposed uses the exact representation of the part and generates an approximate rational description (to within a defined tolerance) of the MAT. The remainder of this paper is structured as follows: Section 2 presents a review of work reported on the construction of MAT. Terminologies used in the paper are defined in Section 3 and Sections 4 and 5 describe the algorithm and present the results of implementation. A discussion on the directions for future work is given in Section 6.

2. Literature review

The MAT was first introduced and explored by Blum [1, 2] to describe biological shape. Soon after its introduction, continuous algorithms for computing the MAT were developed for planar regions. Lee [17] developed an O(n log n) algorithm for polygons with nonconvex corners. His algorithm was based on divide and conquer method where the Voronoi diagram for the polygon is first generated and then the Voronoi edges incident at the nonconvex vertex of the polygon are removed to obtain the MAT for the polygon.

Srinivasan and Nackman [18] presented an O($nh + n$ log n) algorithm for multiply-connected polygons with h holes which is basically an extension of Lee's [17] algorithm. Gursoy and Patrikalakis [5] developed an algorithm to compute the MAT of a multiply-connected planar region bound by circular arcs, and used this algorithm to generate finite element meshes automatically. Their algorithm is based on an offset process directed towards the interior of a shape which is analogous to the propagation of the grass-fire wavefront toward the interior of a shape. The implementation details and extension of the algorithm to trimmed curved surface patches is presented in a companion paper [19].

Sheehy et al. [20] have presented an algorithm for medial surface construction by exploiting the properties of Delaunay triangulation of the domain and also on the basis of definition and classification of the medial surface. The emphasis of the algorithm is on computing the medial surface topology correctly. It is indicated that the developed algorithm can be extended to spheres having contact with the object surface over a finite line or area.

Sherbrooke et al. [12] have presented an algorithm for computing the MAT of 3-D polyhedral objects. The algorithm is based on a classification scheme which relates different parts of the MAT to one another and also provides a continuous representation of the MA and associated radius function. The algorithm is based on offset sweeping of the boundary segments and solving appropriate equations using singular value decomposition technique. They acknowledge that the classification scheme for solids with curved boundaries is more complicated and indicate that it is possible to develop a connectivity scheme similar to the one that they have developed for the polyhedral objects.

Lavender et al. [21] use an octree based approach to determine the Voronoi diagram (more precisely, MAT) for the set theoretic solid models, composed of unions, intersections, and differences of primitive regions represented by a collection of polynomial inequalities.

Kim et al. [11] describe a method for constructing the Voronoi diagram of a simple polygon consisting of points, lines segments and/or arcs which are restricted to lie in a plane. Their algorithm constructs the Voronoi diagram of the interior of a simple polygon using the trisection of plane method. Particular emphasis is placed on the representation of bisectors using rational quadratic Bézier curve which is claimed to unify four different cases.

Constructing the MAT for planar domains with curved boundaries has been studied by Chou [22] and Ramamurthy [23]. The main idea in these approaches revolves around tracing the bisector of pairs of boundary entities and trimming the bisector to obtain the correct topology of the MAT.

Chou [22] has presented an algorithm for constructing Voronoi diagrams (more precisely, MAT) of a domain D, where D is a planar, simply connected, closed shape with a boundary composed of curve segments, including points, line segments, and analytic curves without holes using a differential geometry approach. The boundary curve can have both tangent and curvature discontinuity. It is mentioned that the algorithm can be extended to curves with zero curvature (straight lines) and curves with constant curvature.

The approach outlined by Ramamurthy is based on constructing bisectors of pairs of boundary entities in the domain [13,15]. The approach generates points on the bisector and then interpolates the points to obtain an approximation to the bisector curve [15]. The points on the bisector curve are obtained by constructing a point/curve bisector and then intersecting the bisector with the normal to the curve. Elber and Kim [14] present a symbolic representation scheme for planar bisector curves that allows the construction of bisector curves using B-spline subdivision techniques.

For planar domains with curved entities, the approaches relying on tracing the bisector have to first obtain an approximation of the bisector for each pair of edges (untrimmed bisector for point-curve pair can be constructed [13]) using either a numerical procedure [15] or a combination of symbolic and numerical procedures [14]. Once the bisector segments are obtained these have to be properly trimmed to obtain the correct topology of the MAT. This requires utilities for computing intersections of bisector segments represented as rational curves.

In this paper, an algorithm for generating an approximation of the MAT for 2-D objects with free-form entities has been described. As the input is the exact boundary representation, the approach presented can be categorized as

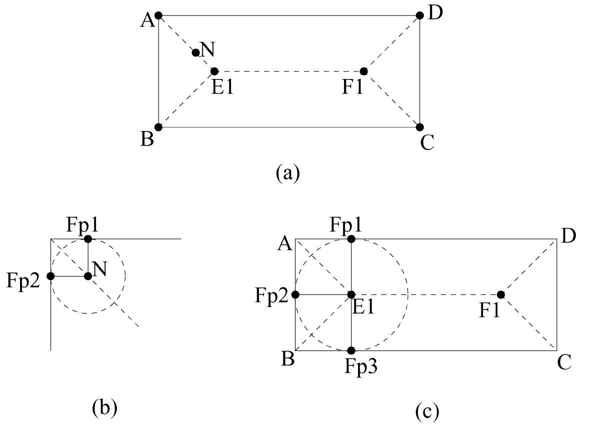


Fig. 4. Classification of points on MA.

a continuous approach. The algorithm is broadly based on the idea of tracing proposed by Chou [22]. The algorithm presented, however, traces the MAT of the part, segment by segment by marching along the boundary (as opposed to stepping along the bisector as done by Chou [22]). This allows the proposed algorithm to handle multiply-connected objects.

3. Preliminaries

Points on the MAT can be classified based on the properties of their maximal disks [24]. A point whose maximal disc touches exactly two separate boundary segments is called *normal point*. Point N (or any point on the line segment (A,E1) excluding the end points A and E1) in Fig. 4(a) is a normal point. Its underlying maximal disk is shown in Fig. 4(b).

A point whose maximal disc touches the domain boundary in three or more separate segments is called *branch point*. Points E1 and F1 in Fig. 4(a) are branch points. Fig. 4(c) shows the maximal disk corresponding to the branch point E1.

A point whose maximal disc touches the boundary in exactly one contiguous set is called an *end point*. Fig. 4(a), shows the end points A, B, C and D. These points touch the boundary at a point and the corresponding maximal disc is of radius zero.

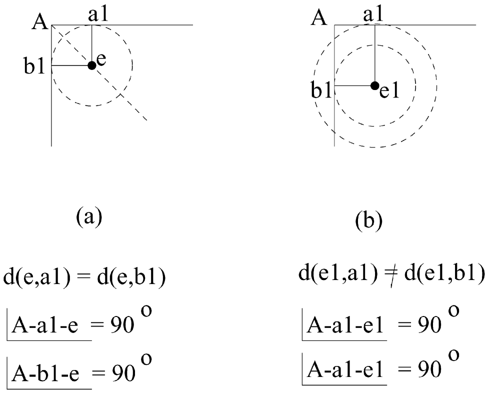


Fig. 5. Distance criterion.

Blum and Nagel [24] view the MAT of a domain as collection of *simplified segments*. A simplified segment is a set of contiguous normal points bound by either a branch point or an end point [24].

A point of contact with the domain boundary, of the underlying disk of a point on the MAT is called the *footpoint* of the point on the MAT. From the definition of the point types in a MAT, a normal point will have two footpoints (Fp1 and Fp2 in Fig. 4(b)), a branch point will have three or more (Fp1, Fp2, and Fp3 in Fig. 4(c)) and an end point will have one or more footpoints.

From the definition of the MAT the following two conditions that have to be satisfied by points on a simplified segment can be derived [22].

Distance criterion. Any point on a simplified segment (apart from the end points of the segment) should be equidistant to two different boundary segments.

Curvature criterion. The radius of curvature of the disk at any MAT point should be less than or equal to the minimum of the local radius of curvature of the boundary segments. Otherwise, the disk will not satisfy the *maximal* disk criterion [24] and so the point is no longer a MAT point even though the equidistant criterion is satisfied (one can call such points as bisector points since those points have to be only equidistant [14]). Curvature criterion becomes very important in the case of determining MAT points for free form entities. Without this condition, the generated points belong to a bisector segment and not to a MAT segment. This is also one reason why any approach based on generating bisectors of free-form edge pairs would require additional processing.

In Fig. 5(a) line segments e–a1 and e–b1 are normal to the boundary segments and also equal in length. So the *point e* is a point on the MAT. But for the *point e1* shown in Fig. 5(b), even though the line segments e1–a1 and e1–b1 are normal to the boundary segments, they are not equal in length. So point e1 is not a point on MAT.

Fig. 6 shows the curvature criterion (the distance criterion is satisfied). Fig. 6(a) shows that the ball is completely inside the boundary segments which implies that the indicated curvature criterion is fully satisfied and point M becomes a MAT point. Violation of this criterion indicates that the ball does not lie inside the boundary segments completely. The point m in Fig. 6(b) is not a MAT point but it still belongs to the set of bisector points. A point will belong to the MAT only when it satisfies both criteria.

4. Overview of algorithm

The algorithm works by finding the footpoint of a normal MAT point on one edge given the footpoint on another edge. This is accomplished by solving the intersection of the

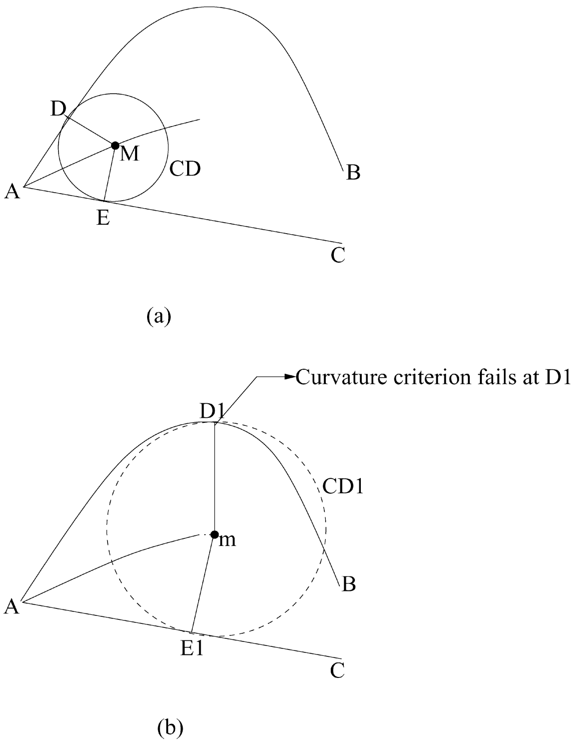


Fig. 6. Curvature criterion.

normals to the two edges and imposing the distance criterion. The next point on the MAT is obtained by finding the intersection of the normals to the edges on the boundary followed by check against curvature criterion. The algorithm therefore replaces the intersection of bisectors (favored by current art [22,23,25]) with intersection of normals. Given that the normals are linear in contrast to the bisectors which can be rational curves or quadratic conics, the computational effort and complexity are reduced considerably.

Tracing of the MAT starts from a convex vertex, where both the footpoints are known (they are the convex vertex itself). It is assumed that the input domain contains atleast one convex vertex. During the tracing of normal MAT points, at chosen intervals, the algorithm checks for the existence of a branch point. This check is a distance check with all other edges in the object. However, the algorithm does not check against all the edges and proposes a more restricted check that gives the correct results. For determining the presence of a branch point it will be shown that it is sufficient to check for interference with edges that have not been traced thus far. This ensures that as the generation of MAT progresses, the check for branch point involves fewer edges.

Once the existence of a branch point is confirmed, two of the footpoints corresponding to the branch point are known and the remaining footpoints of the branch point are determined from the edges that fail the distance check.

Failure of the curvature criterion indicates the presence of a point with locally maximal positive curvature (LMPC) (curvature at a point ‘p’ on a curve is locally maximal if it is greater than the curvatures at points p + ϵ and p − ϵ [22]) in the vicinity. In such cases the MAT segment will terminate at the center of curvature corresponding to the point that has the maximal curvature locally. The tracing procedure used therefore originates at a convex vertex and ends at either a convex vertex or at the center of curvature of a point that has LMPC. At each branch point one or more new segments become available to be traced and the algorithm traces each of these.

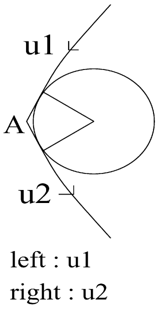


Fig. 7. Parameterization of edges.

The algorithm terminates when all the convex vertices in the part have been visited and no edge pairs remain to be traced at any branch point. The algorithm uses two step sizes. One is used for generating the next MAT point and the other is used while checking for potential branch points.

The output of the algorithm is a set of points on each MAT segment. Rational curves are fit to points in each segment to a desired tolerance to obtain the MAT segments.

5. Algorithm details

The algorithm for constructing the MAT consists of the following steps:

- tracing step;
- checking for a branch point;
- proceeding from a branch point;
- handling a reflex corner;
- termination of the algorithm;
- illustration of the algorithm.

5.1. Tracing step

The boundary segments are parameterized such that the interior (material side) is always to the left of increasing direction of the parameter. In the following, with reference to the convex corner A (refer Fig. 7), the parameter for the left boundary segment is termed as u1 and that for the right is termed as u2. The tracing step consists of the following tasks:

1. Commence tracing from a convex vertex (let e1, e2

be the left and right edge at the vertex and let u1 and u2 be the parameters corresponding to the two edges, respectively).

2. Locate the next footpoint on e2 by incrementing u2 by the tracing step size (δu_2).
3. Find the corresponding footpoint on e1. This step requires the intersection of normals to the two edges and enforcing the distance criterion. The computational effort is in finding the roots of a nonlinear equation in one variable (see Appendix A for details). Convergence is fast as a good initial guess is always available (the previous footpoint on the edge).
4. Check if the intersection point satisfies the curvature check.

The intersection of normals that also satisfies the distance criterion is the MAT point. Such a point exists as the MAT point is the intersection of the bisector and the normals from the respective edges. Here one normal is fixed and the other is searched for such that their intersection is the MAT point (equidistant from the respective edges).

The check against the curvature criterion is a comparison of the radius function (distance between the footpoint and the intersection point) with the radius of curvature at the footpoint. The procedure followed when the curvature criterion is violated at a point during tracing is discussed later during the description of tracing from a branch point.

5.2. Check for branch point

A simplified MAT segment terminates at one of the following—a convex vertex, a branch point, the center of curvature of a point with LMPC. Termination of the tracing of a single MAT segment at a convex vertex is straightforward as it only involves comparing the parameter values with the bounds on the same and checking if the vertex is a convex vertex or not. When the segment terminates at a branch point, however, additional checks are required to first determine the existence of the branch point and then to trace a new segment from the branch point. Termination at the center of curvature of a point with LMPC is indicated by either failure in the curvature criterion or when the end of an edge that is not a convex vertex is reached without encountering a branch point. Both these cases are described later.

The existence of a branch point can be identified by finding the distance of the current MAT point from the remaining boundary segments and checking if this distance is less than the radius function at the current MAT point. It is extremely costly to check the distance at each and every point generated during the tracing with all the remaining boundary segments. Two things are done to reduce the effort in this step. First, the distance check is done for only a few boundary segments (and not all boundary segments in the part). Secondly, the check for existence of a branch point is done only after a finite portion of the MAT segment is generated (and not at every point on the MAT).

The distance check to identify a branch point need not include all the boundary segments because of the following propositions.

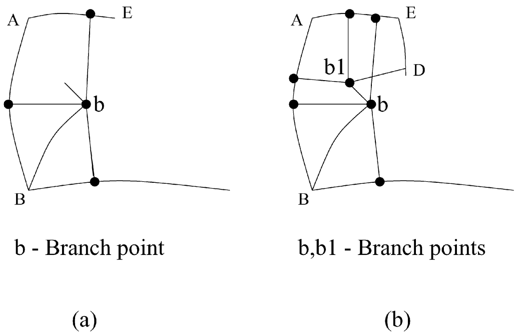
Proposition 1. *Portions of the boundary segments for which the MAT segments have already been traced need not be considered in performing distance check to identify branch points, while generating subsequent MAT segments.*

Proof. Let the proposition be not true. Then at some point 'P' on the MAT segment being generated using the edge pair e_i and e_j, the maximal disk will cut one or more boundary edges that have been already traced. In such a case it is possible to find a smaller disk centered at P that is tangential to an edge that has been cut (e_k) and one of the two edges e_i and e_j in the current pair. This would result in a point on the edge segment cut (e_k) having two MAT points corresponding to it. This violates the onto correspondence between a point on the boundary and its corresponding MAT point (the correspondence relation between the two is a continuous, one-to-one onto map) [22]. The onto correspondence holds even in the presence of reflex (concave) corners when the reflex corner is represented by multiple segments of zero length but with a normal. The maximal disk at point P therefore cannot cut any of the boundary edges that have already been traced. The proposition follows. □

Checking for the existence of a branch point is not done at every point traced on the MAT. It is therefore possible to overshoot a branch point during the tracing procedure. A simple bisection procedure implemented between the last point on the MAT where the distance check was not violated and the current point where the check is violated will identify the branch point.

5.3. Tracing from a branch point

At a branch point, the current segment being traced will continue along one or more branches. For each branch chosen (represented by a pair of edges or an edge and a reflex vertex), the edge common with the edge pair corresponding to the current segment will continue to be traversed in the same direction. The direction of traversal for the other edge (handling reflex vertices will be described later) is opposite to the direction of traversal of the first edge in the pair. If the first edge is traversed along the direction of parameterization (increasing value of its parameter), then the other edge will be traversed in the decreasing value of its parameter and, *vice versa*. The points on the MAT segment corresponding to this pair is determined as described earlier. This process can be made more efficient by noting that when a pair is traversed from a branch point, further checks for branch points need not be made if the pair has a common



b - Branch point

b,b1 - Branch points

(a)

(b)

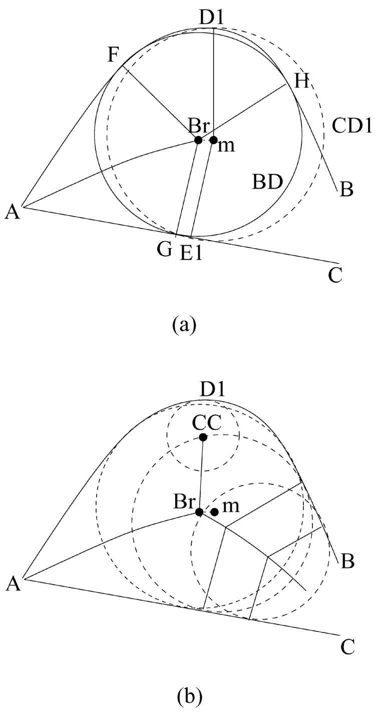
Fig. 8. Existence of a branch point.

convex vertex (Proposition 2). There may be branch points due to the edges in the current pair, however, these will be trapped by the failure of the curvature condition.

The above procedure is repeated recursively for any branch points encountered (where possible) during tracing from a branch point. The recursion terminates at a convex vertex or at the center of curvature of LMPC or when no further branch points are obtained.

Proposition 2. *When a MAT segment is being generated from a branch point by traversing along a pair of edges that share a convex corner, no other branch point* (*with a boundary segment other than the current pair*) *will be encountered during the traversal till the convex corner.*

Proof. The proof is established by contradiction. Fig. 8 shows the tracing of the MAT from a branch point (point ‘b’) using the edge pair (AB, AE) that intersects in a convex corner A. Since they intersect in a convex corner, it is clear that there cannot be any other boundary segment between them. Assume that another branch point (‘b1’) is encountered during the tracing along this edge pair. This implies that there are two MAT segments for the pair (AB, AE). One from the branch point ‘b’ to branch point ‘b1’ and the other that starts/ends from the end point A. As there can be only one MAT segment for a pair of boundary segments (because of the one-to-one onto correspondence between the points on the MAT and their corresponding points on the boundary segments), the tracing of MAT from branch point ‘b’ to the convex corner cannot encounter any other branch point. The proposition follows. □



(a)

(b)

Fig. 9. Violation of curvature condition.

This implies that there is no need for the distance check for the MAT points generated during traversal from a branch point along a pair of edges that share a convex corner. There may be branch points due to the edges in the current pair. These will be, however, trapped by the failure of the curvature condition.

Failure in the curvature criteria indicates that the tracing of MAT has missed a branch point. This is because the check for branch points does not include the current edge pair and in this case, the third footpoint also belongs to one of the edges in the edge pair used for tracing. In Fig. 9(a), MAT corresponding to the edge pair (AC, AB) is being traced. The curvature criterion is violated at the point D1 on edge AB indicating a branch point earlier. The footpoints corresponding to this branch point (Br in the figure) are G, F and H where F, H belong to the same edge AB. In this case, tracing with the current edge pair is stopped and tracing of the MAT commences at the point where the curvature criterion is violated. Portions of the edge to either side of this point are treated as the edge pair. At the point itself the point on the MAT is the center of curvature at that point. Tracing of the MAT with the above edge pair continues till it meets the MAT segment traced when the curvature criterion was violated. Referring to Fig. 9(b), the point on the MAT corresponding to point D1 (where the curvature criterion was violated) is the center of curvature at D1 (point CC in the figure). Treating D_1A and D_1B as the edge pair, the MAT is traced from point CC till it meets the MAT traced earlier at the point ‘Br’. Further tracing of the MAT will now proceed with the edge pair AC and AB corresponding to the footpoints ‘G’ and ‘H’.

Checking against the curvature criterion is better than including the current edge pair also in the check for branch point because, in the latter case tracing of the MAT from the branch point becomes cumbersome. In Fig. 9(a), the distance check that includes the current edge pair (AB–AC) can correctly identify the branch point ‘Br’ and the three footpoints ‘G’, ‘F’ and ‘H’. However, for generating the portion of the MAT corresponding to the segment FH on the edge AB, the termination point ‘CC’ of the corresponding segment of the MAT has to be identified apriori. This will be the center of curvature of the point that has LMPC [22]. The check against the curvature criterion mentioned

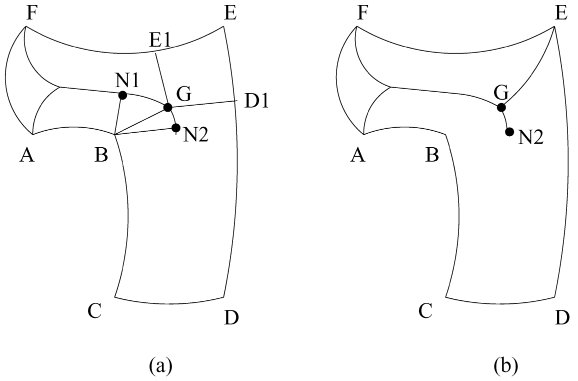


Fig. 10. Handling a reflex corner.

above will identify the termination point correctly enabling the tracing of this portion of the MAT. This algorithm finds the point with LMPC and its center of curvature as required and does not need to process the boundary of the entire domain to identify such points as is done by Chou [22].

5.4. Handling a reflex corner

A reflex corner is represented as a set of circular arcs with zero radius. Associated with each circular arc segment is a normal. The normals are assigned to these segments in order so that the change in the normal while crossing the reflex corner is divided across these segments (see Fig. 10). Tracing of points on the MAT for a pair consisting of an edge and a reflex corner proceeds in the same manner as for tracing the MAT for a pair of edges. The check for the presence of a branch point has to be done here as well as it is possible to encounter a branch point as shown in Fig. 10(a) (Point G is one such point). Fig. 10(b) shows the MAT segments that emanate from branch point G.

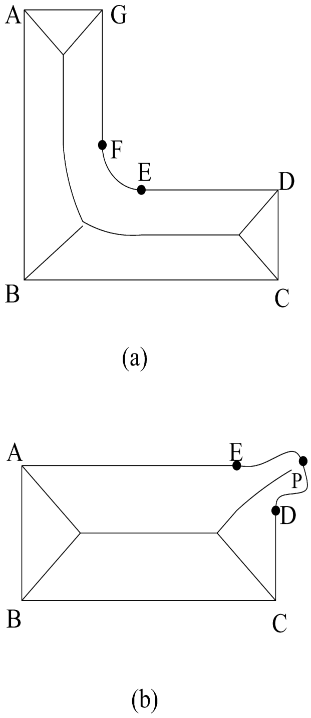


Fig. 11. Tangential edges.

5.5. Handling smooth junctions between edges

The domain may contain edges that are tangential, that is the junction between two edges is smooth. In such cases, the tracing procedure will not terminate when the smooth junction (end of the edge) is reached as it is not a convex vertex. The tracing procedure continues with the neighboring tangential edge and the other edge in the current pair (Fig. 11(a)). When the neighboring tangential edge is the same as the other edge in the current pair (refer Fig. 11(b), where DE is the edge), it indicates that there is a point on this edge where the curvature is locally maximal. In Fig. 11(b), 'L' is the point with LMPC. The MAT segment will now be traced from the last MAT point traced to the center of curvature at this point (point 'P' in Fig. 11(b)). In other cases, termination of a MAT segment will happen at a branch point or convex corner as described earlier.

5.6. Termination of the algorithm

The algorithm ends when all the convex vertices have been visited by the tracing step and no edge pairs remain to be traced at any branch point.

5.7. Illustration of the algorithm

In this section we use a simple example (Fig. 12) to illustrate the procedure to construct the MAT. Tracing starts at a convex vertex, 'A' (Fig. 12(a)). For every foot point on edge AB (obtained by incrementing the parameter by the tracing step size) the tracing step computes the corresponding foot point on edge AG and the point on the MAT, P_i. In Fig. 12(b) the distance criterion is violated at point P' as the disk corresponding to this point cuts the edge FG. The procedure then uses the bisection method between this point and the previous point found on the MAT to identify the branch point B1 (Fig. 12(c)). The boundary segments corresponding to the branch point are AB, AG and FG. Tracing now continues with the pair AG and FG. As these two edges share a convex vertex (G) no check for branch point is required and the tracing of the MAT segment ends at G (Fig. 12(d)). The tracing procedure next starts at B1 with the other possible pair of edges, namely AB and FG (Fig. 12(e)).

Tracing with this pair continues till the point N1 on the MAT where one of the footpoint (for the edge AB) reaches the limiting value (Fig. 12(e)). This end point is also a concave vertex that has been represented as a sequence of zero length edges with normals uniformly varying between the normals to edges AB and BC at B. The procedure now traces with pairs formed by the edge FG and each of these zero length edges at B in order to obtain the segment N1–

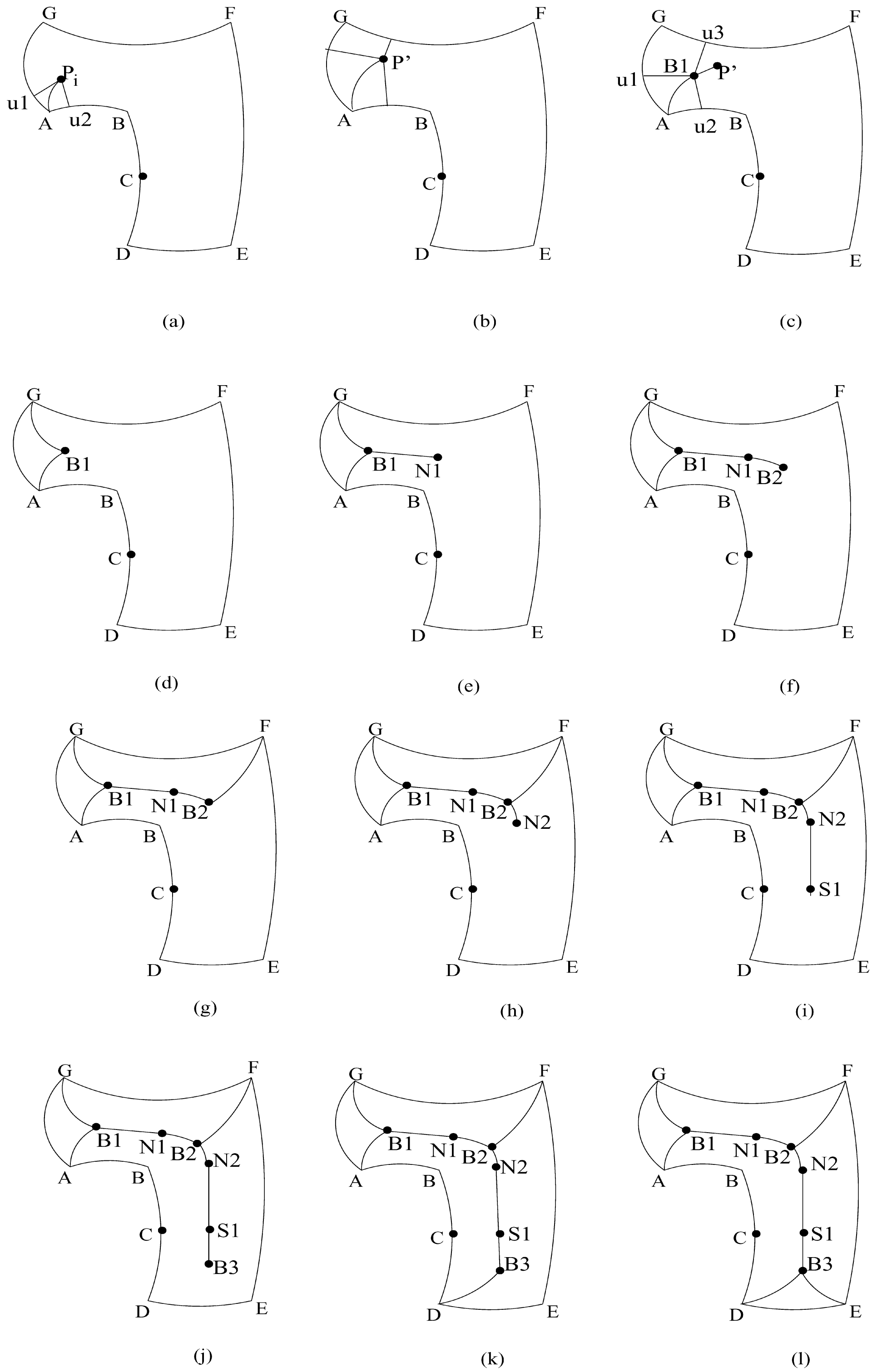


Fig. 12. Illustration of the algorithm.

B2 of the MAT (Fig. 12(f)). As explained earlier, B2 is identified as the branch point. The procedure now traces the segment B2–F of the MAT using the edge pair FG and EF (Fig. 12(g)). Tracing starts again from the branch point B2 and proceeds with the remaining dummy edges representing the concave vertex B and the edge EF. After the last of the segments in the list of dummy edges is reached (at the point N2 on the MAT in Fig. 12(h)) the tracing continues with the edge pair BC and EF. Tracing of this pair stops when BC reaches the limiting parameter value (at the MAT point S1 in Fig. 12(i)). Since C is a smooth corner, tracing of MAT continues between the pair CD and EF. Tracing with this

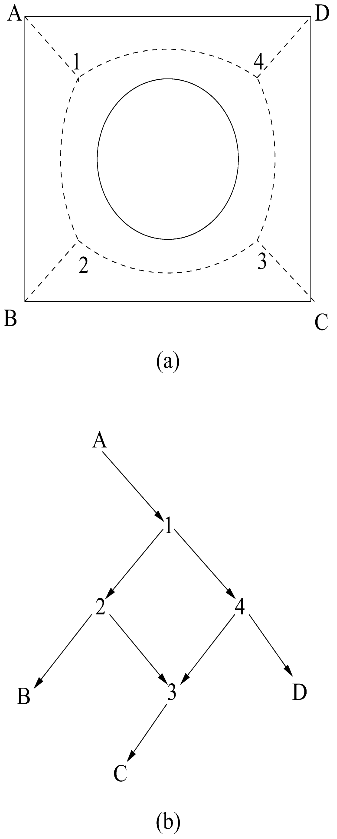


Fig. 13. Representation of MAT.

pair stops with the detection of the branch point B3 (Fig. 12(j)) and the MAT segment B3–D is traced using the edge pair ED–DC (Fig. 12(k)).

The procedure again returns to the branch point B3 to trace with the other possible edge pair DE–EF (Fig. 12(l)). When the tracing with this edge pair ends at point E, there are no edge pairs at any branch point left to be traced and there are no convex vertices that have not been visited. The procedure therefore terminates (Fig. 12(l)).

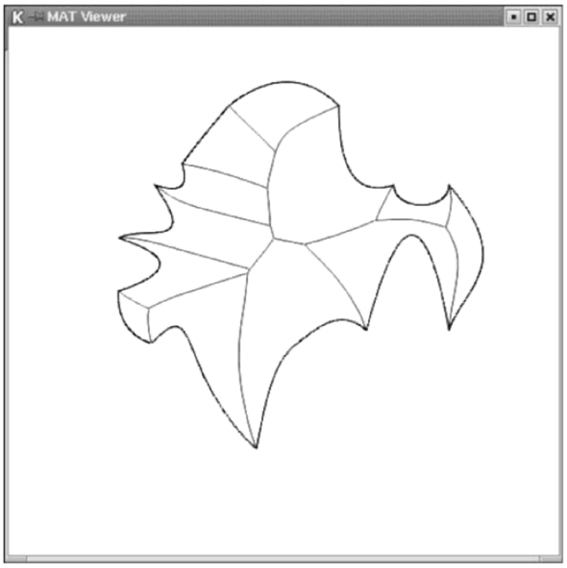


Fig. 14. MAT for test object 1.

6. Results and discussion

The algorithm described has been implemented and this section presents the results obtained for some typical planar domains. The input to the algorithm is the B-Rep of the object. Curved entities are represented using rational B-Spline curves. Output is the set of MAT segments and each MAT segment is available as either a set of points or set of B-Spline segments.

The step size for tracing is chosen to be smaller than the smallest feature in the domain (tracked through the length of boundary segments) so that no features are missed. Presently the implementation uses a step size of 0.001 (for domains scaled to 1) or one hundredth of the smallest edge whichever is smaller, for tracing. This choice gives a fine approximation to the MAT with reasonably real time response (order of seconds for the parts shown in this section). A larger step size will result in a coarser approximation. The step size for checking for branch points is taken to be an order of magnitude greater than the step size for tracing.

Since the algorithm traces the MAT segment by segment, and also each segment starts from a branch point (except the starting one) with a predetermined direction (parameter in the edge to the left of MAT always decreases and that for the edge to the right of the MAT always increases), directed graphs can be used to represent MAT. The graph representing the MAT will be acyclic for multiply connected domains. Fig. 13 shows the MAT for a multiply-connected object and its representation as a directed acyclic graph(dag). The first node represents the starting convex corner and the end nodes (no connecting segment from these nodes) represent the remaining convex corners and centers of curvature of points with LMPC, and the interior nodes represent the branch points.

MAT obtained for some typical free-form planar objects are shown in Figs. 14–19. The algorithm is able to generate the MAT for objects with complex boundary profiles (Figs. 14 and 15), concave vertices (Fig. 16), spiral shaped (Fig. 17), and, multiply-connected objects (Fig. 18) and objects with smooth corners (Fig. 19). Smooth vertices are marked by dots to distinguish the tangential edges in Fig. 19. Note that the MAT is generated even for domains with constricted passages and sharp changes in the curvature along the boundary (Fig. 15).

Table 1 shows the time taken for generating MAT for some of the figures. The implementation is on a PIII machine with 256MB RAM. The step size for tracing and checking for branch points are 0.001 and 0.01, respectively. If the step size is larger, then a coarser MAT will be generated but takes lesser time. Figs. 20 and 21 show the MAT for the object in Fig. 15 with step sizes for tracing set

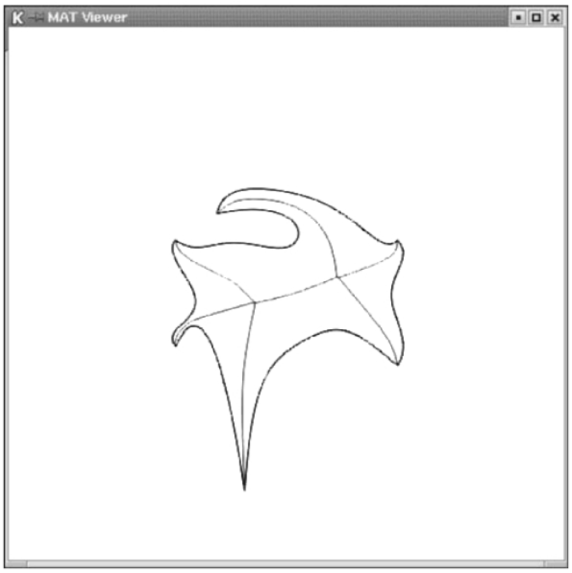


Fig. 15. MAT for test object 2.

as 0.01 and 0.1, respectively. Table 2 shows the time taken for generating MAT with different step sizes for the object shown in Fig. 15.

6.1. Discussion

The completeness of the algorithm can be derived from the graph structure of MAT. 'Completeness' here refers to the generation of all the segments constituting the MAT. The starting and end nodes in the graph representing the MAT correspond to convex corners and centers of curvature of points with LMPC present in the object. Each branch of the graph corresponds to a MAT segment. Each downward traversal of a branch corresponds to the generation of a MAT segment. The algorithm starts at A (Fig. 13(b)) and proceeds toward 1 generating the segment A–1. From 1 the downward traversal generates the MAT segments 1–2 and 1–4. Then it starts form 2 and generates 2–B and 2–3. From 4, MAT segments 4–3 and 4–D are generated. Finally 3–C is generated. A topological sort [27] (a variant of the depth first search) will list all the convex corners in the object indicating that all the MAT segments have been generated.

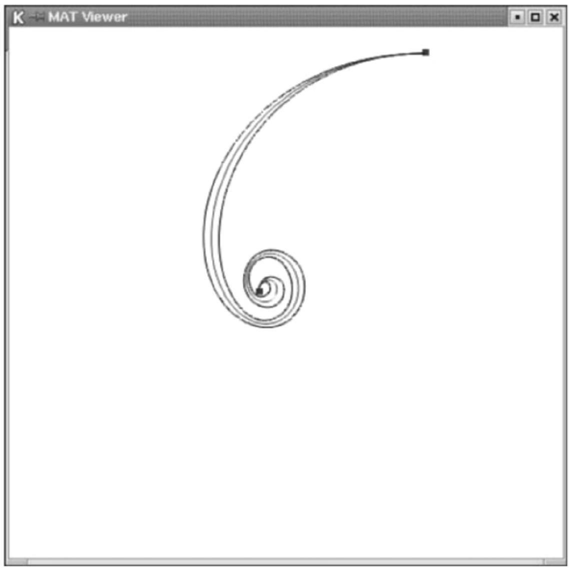


Fig. 17. MAT for spiral shaped object.

The proposed algorithm does not trace the bisectors for all the boundary pairs and then trim the bisectors. Rather it traces the MAT segments directly by tracing the boundary of the domain. As the points on the MAT determined are based on the definition of the MAT, the result will always be correct. Branching from one pair of boundary segments to another is also based on the

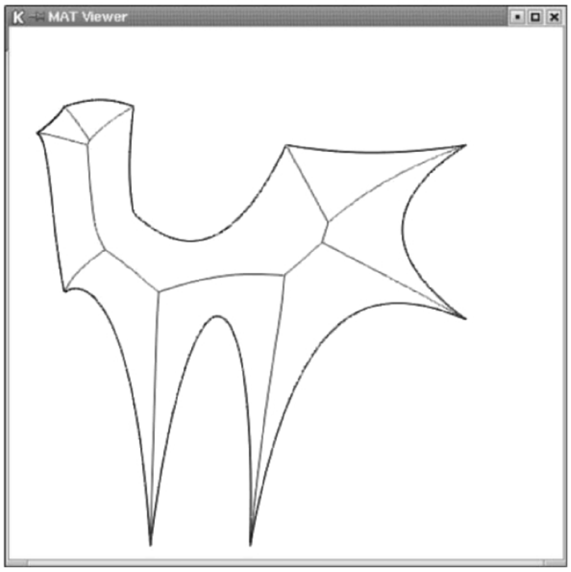


Fig. 16. MAT for test object 3.

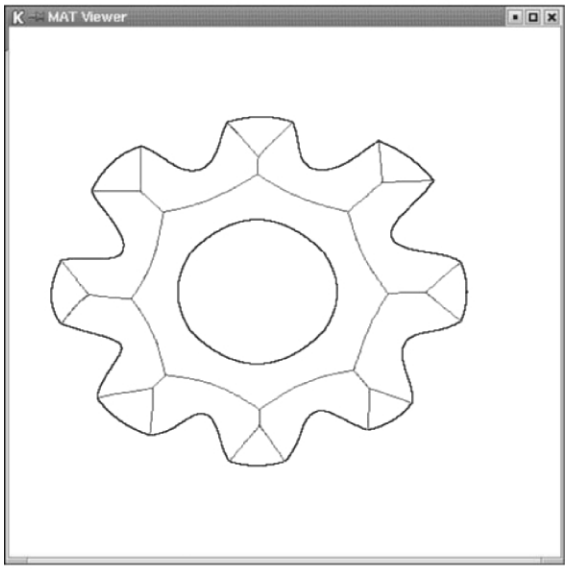


Fig. 18. MAT for a multiply connected object.

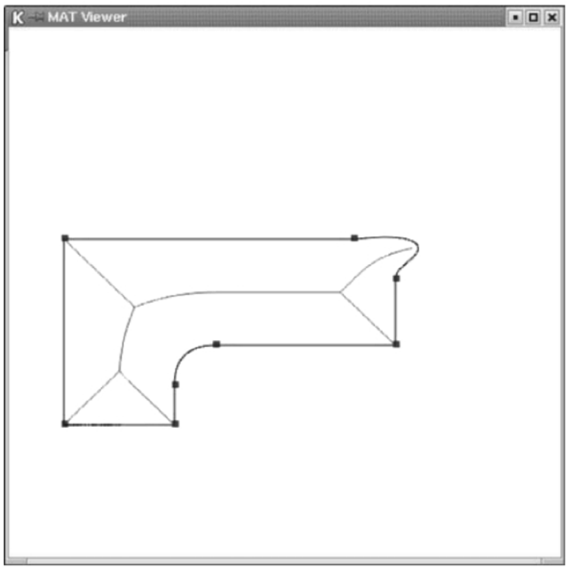


Fig. 19. MAT for an object having tangential edges.

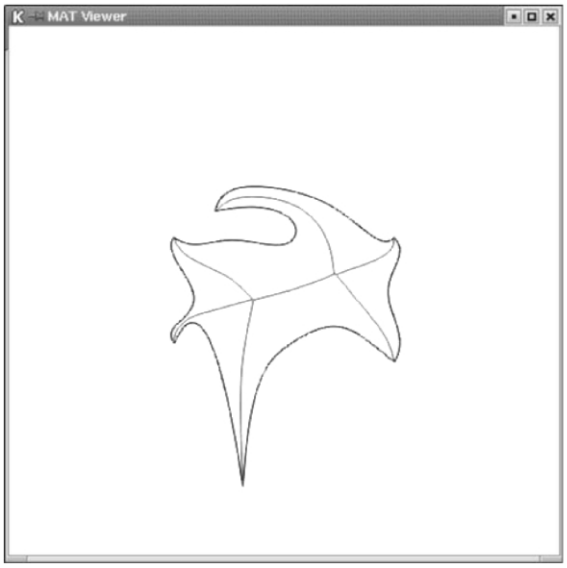


Fig. 20. MAT for test object 2 with step size 0.01.

continuity of the MAT. The graph structure used for both tracing the boundary and storing the MAT ensures completeness. As mentioned earlier, numerical tracing of bisectors cannot be avoided and therefore the proposed technique will be as accurate as those based on tracing bisectors. It is also believed that the trimming of bisectors will require as much computation if not more as generating them to obtain the MAT [15]. Also, the techniques based on tracing the bisectors will have to subject the MAT segments generated to the curvature criterion [22]. The algorithm does not have the problems of degeneracy as mentioned by Kim et al. [11]. Degeneracy refers to the need for processing many branches in the algorithm, each dealing with an exception to the main algorithm [11]. The main steps of the algorithm described are those of tracing the simplified segment, checking for a branch point and handling reflex corners. Each step in itself has no exceptions to handle and works for the general case. Let n be the number of boundary segments and N be the number of points on each boundary segments. The complexity of the main computational tasks in the procedure are O(1) for computing the intersection of two normals and $O(n)$ for implementing the distance check. The complexity for finding the part of the MAT corresponding to a single boundary edge is $O(Nn)$ and for the complete domain is $O(Nn^2)$ which is the worst case complexity of the algorithm.

Table 1
Time taken for generation of MAT for typical objects

Figure no.	No. of MAT seg.	Time (s)
14	21	3
15	9	2
16	15	1
17	32	3

A formal analysis of the algorithm with respect to the effect of numerical errors due to finite precision computation has not been done. It is believed that errors in the computation will not affect the topology of the resulting MAT (and therefore result in abnormal behaviour of the algorithm). It must be mentioned that as long as the feature size constraint is not violated, even a coarser step size for tracing will yield the correct connectivity between the MAT segments. The algorithm is robust enough to capture an overshot branch point. The branching decisions are therefore made correctly, while maintaining continuity of both the tracing procedure and of the MAT.

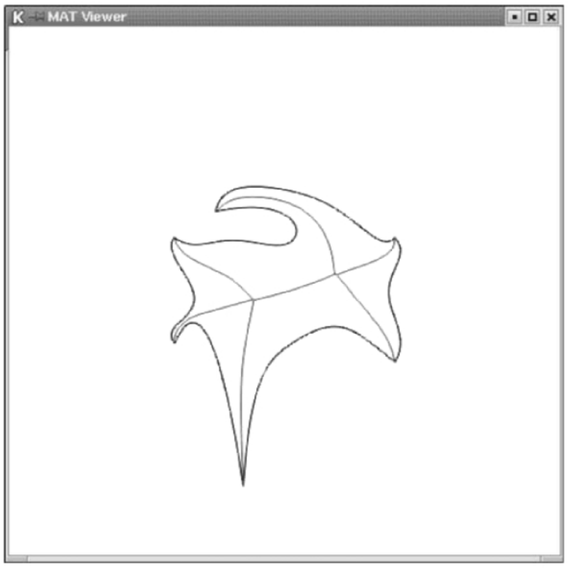


Fig. 21. MAT for test object 2 with step size 0.1.

Table 2
Time taken for generation of MAT for various step sizes

Figure no.	Step size	Time (s)
15	0.001	2
20	0.01	1
21	0.1	1

7. Conclusions

An algorithm for generating the MAT for planar domains with free form boundaries has been described. The algorithm generates the MAT by a tracing technique that marches along the object boundary rather than the bisectors of the boundary entities. Criteria derived from the definition of the MAT are used to generate the points on the MAT and identify the branch points. The algorithm is shown to be robust and correct. A smooth approximation to the desired accuracy is obtained by interpolating the points obtained on each simplified segment of the MAT. Results of implementation on typical domains, including multiply-connected domains have been provided.

Appendix A. Determining point on MA

The procedure outlined here for determining the footpoint on the other edge and the intersection of the normals to obtain the point on MA is based on the derivation in Ref. [14]. The procedure is described here using the notation used in this paper for completeness.

Let $C_1(u_1) = (x_1(u_1), y_1(u_1))$ and $C_2(u_2) = (x_2(u_2), y_2(u_2))$ be the underlying curve of edges e_1 and e_2, respectively. Curves $C_1(u_1)$ and $C_2(u_2)$ are planar regular rational curves with C^1-continuity.

Let $N_1(u_1) = (-y_1'(u_1), x_1'(u_1))$ and $N_2(u_2) = (-y_2'(u_2), x_2'(u_2))$ be the (in plane) normals of $C_1(u_1)$ and $C_2(u_2)$, respectively. The intersection point of the normal lines is given by

$$C_1(u_1) + N_1(u_1)\alpha = C_2(u_2) + N_2(u_2)\beta$$

for some α and β. We then have the following two equations in two unknowns, α and β :

$$x_1(u_1) - y_1'(u_1)\alpha = x_2(u_2) - y_2'(u_2)\beta,$$
$$y_1(u_1) - x_1'(u_1)\alpha = y_2(u_2) - x_2'(u_2)\beta.$$

Solving for α and β

$$\alpha = \alpha(u_1, u_2) = \frac{(x_1(u_1) - x_2(u_2))x_2'(u_2) + (y_1(u_1) - y_2(u_2))y_2'(u_2)}{x_2'(u_2)y_1'(u_1) - x_1'(u_1)y_2'(u_2)},$$

$$\beta = \beta(u_1, u_2) = \frac{(x_1(u_1) - x_2(u_2))x_1'(u_1) + (y_1(u_1) - y_2(u_2))y_1'(u_1)}{x_2'(u_2)y_1'(u_1) - x_1'(u_1)y_2'(u_2).} \tag{A1}$$

The point of intersection of the two normals is given by:

$$\begin{aligned} P(u_1, u_2) &= C_1(u_1) + N_1(u_1)\alpha(u_1, u_2) \\ &= C_2(u_2) + N_2(u_2)\beta(u_1, u_2). \end{aligned} \tag{A2}$$

For $P(u_1, u_2)$ to be a point on the MA of the edges e_1 and e_2, it must be equidistant from $e_1(C_1(u_1))$ and $e_2(C_2(u_2))$:

$$\begin{aligned} \|P(u_1, u_2) - C_1(u_1)\| &= \|C_1(u_1) + N_1(u_1)\alpha(u_1, u_2) - C_1(u_1)\| \\ &= \|C_2(u_2) + N_2(u_2)\beta(u_1, u_2) - C_2(u_2)\| \\ &= \|P(u_1, u_2) - C_2(u_2)\| \end{aligned}$$

The above condition simplifies to

$$\|N_1(u_1)\alpha(u_1, u_2)\| = \|N_2(u_2)\beta(u_1, u_2)\|.$$

Squaring the above equation and substituting $\alpha(u_1, u_2)$ and $\beta(u_1, u_2)$ from Eq. (A1) into the resulting equation, we get

$$\begin{aligned} 0 &= \langle N_1(u_1), N_1(u_1)\rangle\alpha(u_1, u_2)^2 - \langle N_2(u_2), N_2(u_2)\rangle\beta(u_1, u_2)^2 \\ &= \langle N_1(u_1), N_1(u_1)\rangle \\ &\quad \times \frac{((x_1(u_1) - x_2(u_2))x_2'(u_2) + (y_1(u_1) - y_2(u_2))y_2'(u_2))^2}{(x_2'(u_2)y_1'(u_1) - x_1'(u_1)y_2'(u_2))^2} \\ &\quad - \langle N_2(u_2), N_2(u_2)\rangle \\ &\quad \times \frac{((x_1(u_1) - x_2(u_2))x_1'(u_1) + (y_1(u_1) - y_2(u_2))y_1'(u_1))^2}{(x_2'(u_2)y_1'(u_1) - x_1'(u_1)y_2'(u_2))^2}. \end{aligned} \tag{A3}$$

The denominator of Eq. (A3) would vanish when the normals of the two curves are either parallel or opposite (the normals overlap) at some values of the parameters u_1 and u_2. For the case where $N_1(u_1)$ and $N_2(u_2)$ are neither parallel nor opposite, the denominator of Eq. (A3) cannot vanish and the condition in Eq. (A3) becomes:

$$\begin{aligned} 0 = F_1(u_1, u_2) &= (x_1'2(u_1) + y_1'2(u_1))\big((x_1(u_1) - x_2(u_2))x_2'(u_2) \\ &\quad + (y_1(u_1) - y_2(u_2))y_2'(u_2)\big)^2 - (x_2'2(u_2) + y_2'2(u_2))\big((x_1(u_1) \\ &\quad - x_2(u_2))x_1'(u_1) + (y_1(u_1) - y_2(u_2))y_1'(u_1)\big)^2. \end{aligned} \tag{A4}$$

Given the parametric value u_1 of the footpoint $C_1(u_1)$ on edge e_1 of the point on the MA, the solution of Eq. (A4) is the parameter u_2 of the corresponding footpoint $C_2(u_2)$ on edge e_2. The point on the MA is obtained from Eq. (A2).

The condition:

$$x_1'(u_1)y_2'(u_2) - x_2'(u_2)y_1'(u_1) = 0,$$

indicates that the normals of the curves are parallel or opposite. For the cases where the normals overlap each other, the parametric variables (u_1 and u_2) must also satisfy the following conditions:

$$(x_1(u_1) - x_2(u_2))x_1'(u_1) + (y_1(u_1) - y_2(u_2))y_1'(u_1) = 0,$$

$$(x_1(u_1) - x_2(u_2))x_2'(u_2) + (y_1(u_1) - y_2(u_2))y_2'(u_2) = 0.$$

In these cases, the footpoint $C_2(u_2)$ is obtained by intersecting the normal at $C_1(u_1)$ with the edge e_2. The mid-point of the vector between the two footpoints $C_1(u_1)$ and $C_2(u_2)$ is taken as the point on the MA.

References

[1] Blum H. A transformation for extracting new descriptors of shape. In: Dunn W, editor. Models for the perception of speech and visual form. Cambridge: MIT Press, 1967. p 362–80.

[2] Blum H. Biological shape and visual science (Part I). J Theor Biol 1973;38:205–87.

[3] Baja GS, Thiel E. (3–4)-Weighted skeleton decomposition for pattern representation and description. Pattern Recogn 1994;27(8):1039–49.

[4] Montanari U. Continuous skeletons from digitized images. J Assoc Comput Machinery 1969;16(4):534–49.

[5] Gursoy HN, Patrikalakis NM. An automatic coarse and finite surface mesh generation scheme based on medial axis transform: Part 1 Algorithms. Engng Computers 1992;8:121–37.

[6] Armstrong CG. Modeling requirements for finite element analysis. Comput Aided Des 1994;26(7):573–8.

[7] Ramanathan M. Medial axis transform for the prediction of shrinkage and distortion in castings. M.Sc. Thesis, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India.

[8] O'Rourke J. Computational Geometry in C. Cambridge; 1993.

[9] Gelston S, Dutta D. Boundary surface recovery from skeleton curves and surfaces. Comput Aided Geometr Des 1995;12:27–51.

[10] Wolter FE. Cut locus and medial axis in global shape interrogation and representation. Comput Aided Geometr Des 1992;.

[11] Kim D, Hwang I, Park B. Representing the Voronoi diagram of a simple polygon using rational quadratic Bèzier curves. CAD 1995; 27(8):605–14.

[12] Sherbrooke EC, Patrikalakis NM, Brisson E. Computation of the medial axis transform of 3-D polyhedra. Proceedings of the Solid Modeling '95. p. 187–99.

[13] Farouki R, Johnstone J. The bisector of a point and a plane parametric curve. Comput aided Geometr des 1994;11(2):117–51.

[14] Elber G, Kim M. Bisector curves for planar rational curves. CAD 1998;30(14):1089–96.

[15] Farouki R, Ramamurthy R. Specified-precision computation of curve/curve bisectors. Int J Comput Geometr Applic 1998;8(5,6):599–617.

[16] Culver T, Keyser J, Manocha D. Accurate computation of the medial axis of a polyhedron. Proceedings of the Symposium on Solid Modeling, Ann Arbor, Michigan; 1999. p. 179–90.

[17] Lee DT. Medial axis transformation of a planar shape. IEEE Trans Pattern Anal Mach Intell 1982;PAMI-4(4):362–9.

[18] Srinivasan V, Nackman LR. Voronoi diagram for multiply-connected polygonal domains I: Algorithm. IBM J Res Develop 1987;31(3): 361–72.

[19] Gursoy HN, Patrikalakis NM. An automatic coarse and finite surface mesh generation scheme based on medial axis transform: Part 2 Implementation. Engng Comput 1992;8:179–96.

[20] Sheehy DJ, Armstrong CG, Robinson DJ. Shape description by medial surface construction. IEEE Trans Visualisat Comput Graphics 1996; 2(1):62–72.

[21] Lavender D, Bowyer A, Davenport J, Wallis A, Woodwark J. Voronoi diagrams of set theoretic solid models. IEEE Comput Graphics Applic 1992;September:69–77.

[22] Chou JJ. Voronoi diagrams for planar shapes. IEEE Comput Graphics Applic 1995;March:52–9.

[23] Ramamurthy R. Voronoi diagrams and medial axes of planar domains with curved boundaries. PhD Thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan; 1998.

[24] Blum H, Nagel RN. Shape description using weighted symmetric axis features. Pattern Recogn 1978;10:167–80.

[25] Held M. Voronoi diagram and offset curves of curvilinear polygons. Comput Aided Des 1998;30(4):287–300.

[26] Ho S-B, Dyer C. Shape smoothing using medial axis properties. IEEE Pattern Anal Mach Intell 1986;8(4):512–9.

[27] Aho AV, Hopcroft JE, Ullman JD. Data structures and algorithms, Reading, MA: Addison-Wesley; 2000.

Ramanathan M. is currently working as a research scholar in the Department of Mechanical Engineering at the Indian Institute of Science, Bangalore, India. He earned a BE degree in Mechanical Engineering from Thiagarajar college of Engineering, Madurai, India and received the Masters' degree from Indian Institute of Science. His current research interests include geometric modeling, computational geometry and computer graphics.

B. Gurumoorthy is currently an Associate Professor in the Centre for Product Design and Manufacturing and the Department of Mechanical Engineering at the Indian Institute of Science in Bangalore, India. He received his B. Tech in Mechanical Engineering from Indian Institute of Technology, Madras in 1982. He received his ME and PhD in Mechanical Engineering from Carnegie Mellon University, Pittsburgh, USA in 1984 and 1987, respectively. His current research interests are in the areas of geometric modelling, features technology, reverse engineering and rapid prototyping