

A tracing algorithm for constructing medial axis transform of 3D objects bound by free-form surfaces

M. Ramanathan *, B. Gurumoorthy
Department of Mechanical Engineering
Indian Institute of Science
Bangalore 560 012, INDIA
Email: bgm@mecheng.iisc.ernet.in

Abstract

This paper presents an algorithm for generating the Medial Axis Transform (MAT) of 3D objects with free-form boundaries. The algorithm proposed uses the exact representation of the part and generates an approximate rational spline description (to within a defined tolerance) of the MAT. The algorithm generates the MAT by a tracing technique that marches along the object boundary. 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 points and the search for these junction points is done in an efficient way. The algorithm works for multiply-connected objects as well. Results of implementation are provided.

Keywords : Medial Axis Transform, Voronoi diagram, Skeleton, free-form boundaries

1. Introduction: Medial Axis Transform

The Medial Axis Transform (MAT) was first introduced by Blum [3] to describe biological shape. Since its introduction, the MAT has found use in a wide variety of applications that primarily involve reasoning about geometry or shape [18].

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 balls (or disks in 2-D) which are maximal in D , together with the limit points of this locus. A closed ball (or disk) is said to be *maximal* in a subset D of the 3D (or 2D) space if it is contained in D but is not a proper subset of any other ball (or disk) 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 ball or disk. The Medial Axis Transform (MAT) of D is the MA together with its associated radius function.

There have been several efforts reported for the construction of MAT for 3D objects. Most of these efforts focus on polyhedral objects. Very few algorithms that can handle free-form entities have been reported. In general, obtaining a continuous description of MAT (in 2D or 3D) has proven to be difficult. The MAT can be constructed from the bisectors (locus of equidistant points) of the entities on the domain boundary. However, a rational representation of the bisector is proven to be available only for certain special cases - between two rational space curves [8] and for CSG primitives in special configurations [11, 7]. Bisector surface between two rational surfaces in R^3 is non-rational in general [11]. Also, the bisector of simple geometries is not always simple. While the bisector of two lines in the plane is a line, the bisector of two skewed lines in R^3 is a hyperbolic paraboloid of one sheet [10]. Moreover, the post-processing involved in identifying valid bisectors and trimming them have proven to be costly even in 2D and this has led to approaches based on tracing and intersection of normals [18]. In 3D, this has led to techniques based on discretisation [21, 19, 2, 6, 5], spatial decomposition [16, 12, 13] and numerical tracing [20, 4].

In general, algorithms that claim to address free-form objects handle these by discretising the non-linear entities into polyhedral convex faces [12], or using point-sets [19] or spatially decompose the object [16]. Using the point-set model instead of exact representation does not yield the correct geometry of the MAT [21] unless optimization techniques are used. When free-form faces are discretised into polyhedral convex faces, it must be noted that additional effort is required in trimming and post-processing the generated MAT to be in conformity with the topology of the original free-form object. The additional MAT segments will distort any reasoning based on MAT (see Figure 1(b), where the MAT is substantially different when the exact representation is used (Figure 1(a))). For finite element applications, such a MAT representation may be sufficient but

* Currently, a Post-Doctoral Associate, Department of Computer Science, Technion, Israel. This work was carried out when he was a Ph.D. student in the Indian Institute of Science.

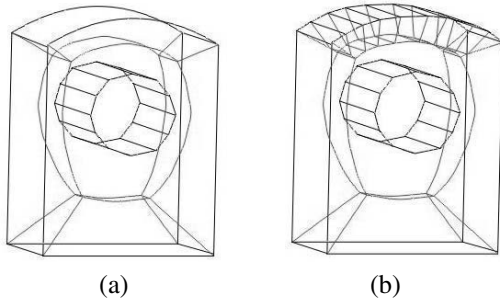


Figure 1. MAT for exact (a) vs. discretised (b) object

exact representation of the part has to be used to construct the MAT in applications where the geometry of the part has to be reconstructed from the MAT [14]. Spatial decomposition methods have shown to work for polyhedral and set-theoretic solids models. Numerical tracing method appear to have been used only for polyhedral solid models.

Using the exact representation of the part for constructing the MAT eliminates the need for additional processing required to eliminate the artificial segments that may arise due to the discretisation of the curved entities into polyhedral entities [20].

To the best of the knowledge of the authors, no algorithm thus far has used the exact representation of a free-form object to generate the MAT. Construction of the Voronoi surfaces of CSG objects has been addressed in [7] as a solution of algebraic equations but trimming of the Voronoi surfaces (to obtain the MAT) is non-trivial. No implementation results are provided.

In this paper, the problem of generating a MAT for solids with free-form curved entities is addressed. It is believed that the richest shape variety can be reached by solids with free-form boundary surfaces that are typically defined by Non-Uniform Rational B-Spline (NURBS) surface patches [22]. The algorithm uses the fact that, when the boundary consists of finite union of rational surface patches, the MAT becomes a semi-algebraic set that can be decomposed into a finite union of smooth curves and surfaces [22].

The proposed algorithm uses the exact representation of the part and generates an approximation (to within a defined tolerance) of the MAT that is represented by rational spline curves and surfaces. The algorithm is approximate in the sense that, though the points that are generated lie on the MAT, the same cannot be said when a point located in between the two generated points is considered. As the input is the exact boundary representation, the approach presented can be categorized as a continuous approach. The algorithm presented traces the MAT of the part, segment by segment by marching along the boundary. This algorithm is an extension of the 2D MAT algorithm presented in [18]. Unlike the 2D case, the equations to be solved here are multi-dimensional, precisely, five equations in six variables. Points on the MAT are obtained by tracing along one

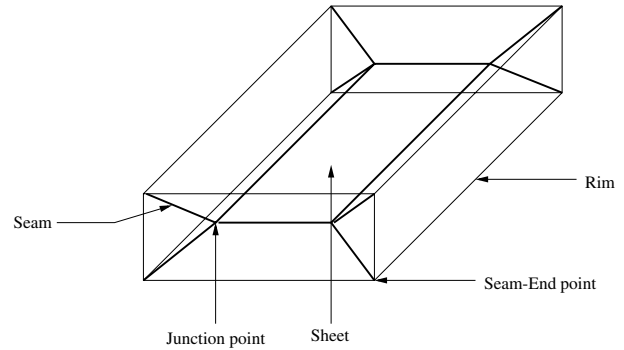


Figure 2. Classification of points on 3D MAT

variable (tracing parameter) and solving the set of five equations for the other variables. Moreover in 3D, the determination of the tracing parameter and the direction of traversal is not obvious as in 2D. An elegant formulation and a procedure to determine the tracing direction are presented. Unlike the approach in [20] which is also based on tracing, the present algorithm traverses along the boundary to identify a point on the seam rather than traversing along the tangent to a point on the seam. This facilitates the handling of free-form boundary entities unlike in [20] where only polyhedral entities have been handled. This is because we do not need to determine the tangent to the MAT entities at every step.

The remainder of this paper is structured as follows: Next section presents some preliminaries that are necessary for describing the algorithm. Overview of the algorithm is presented in the following section. Details of the steps used are presented next followed by results of implementation and a discussion of the results and the algorithm.

2. Preliminaries

In this paper, the term ‘boundary segments’ refers to boundary surfaces (a free-form surface represented in Bézier or B-spline form), reflex edges as well as reflex corners in an object.

2.1. Classification of points in 3D MAT

The various elements that generally comprise the 3D MAT have been defined by Sherbrooke et. al. [20] and Turkiyyah *et al.* [21] (a formal classification can also be found in [15]). These are reproduced here for convenience and are illustrated in Figure 2.

Touchpoint : A point at which a maximal sphere is tangent to the boundary surface.

Seam : A connected space curve consisting of points that have three touchpoints and are non-manifold. Points on the seam are called *seam points*.

Junction point : A point where the seams intersect.

Seam-End point : These points do not arise from the maximal ball condition of the MAT, but are actually the

limit points of the MAT. A seam-end point generally results when a seam runs into the boundary of the solid. Therefore, vertices with convex edges incident are seam-end points [20].

Skeletal edge : A connected space curve consisting of points whose maximal spheres have a single touchpoint with the surface of the object. A skeletal edge is possible only if a surface contains principal curve [21].

Sheet : Sheet is a manifold subset of the MAT which is maximal in the sense that it is connected and bounded only by seams and skeletal edges. A seam corresponds to the intersection of sheets.

Sheet point : A point in the interior of a sheet. Sheet points have exactly two touchpoints.

Sheet-End point : These points also are among the limit points of the MAT, but whereas a seam-end point is a limit point of seam, a sheet-end point is a limit point of a sheet. All seam points are sheet-end points but all sheet-end points need not be seam points.

Rim : Connected components of sheet-end points are called rims. Convex edges, resulting from the intersection of sheets with the boundary are rims.

From the classification of the points on 3D MAT, any seam point should be equidistant to three boundary segments, any sheet point should be equidistant to two boundary segments and the junction point should be equidistant to four or more boundary segments.

2.2. Conditions that determine a point on MAT

From the definition of the MAT the following two conditions that have to be satisfied by points on a MAT segment can be derived[18].

Distance criterion : Any point on a seam (apart from the other points of the MAT) should be equidistant to three different boundary segments. This is equivalent to saying that any point on the simplified segment of MAT in 2D (apart from the terminal points) should be equidistant to two boundary segments [18].

Curvature criterion : In 2D, for free-form boundaries, it is shown that[18] 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. In 3D, the radius of curvature of the ball (or sphere) at any point on MAT should be less than or equal to the minimum of the radius of curvatures (reciprocal of the maximum normal curvatures) at the touchpoints of the boundary . Otherwise, the ball will not satisfy the *maximal* ball criterion and so the point is no longer a MAT point even though the distance criterion is satisfied (one can call such points as bisector points since these points are equidistant [8] from the boundary entities). Curvature criterion becomes very important in the case of determining MAT points for free-form entities [18]. Without this condition, the generated points belong to a bisector segment and not to a MAT segment.

In Figure 3(a) line segments e-a1, e-b1 and e-c1 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

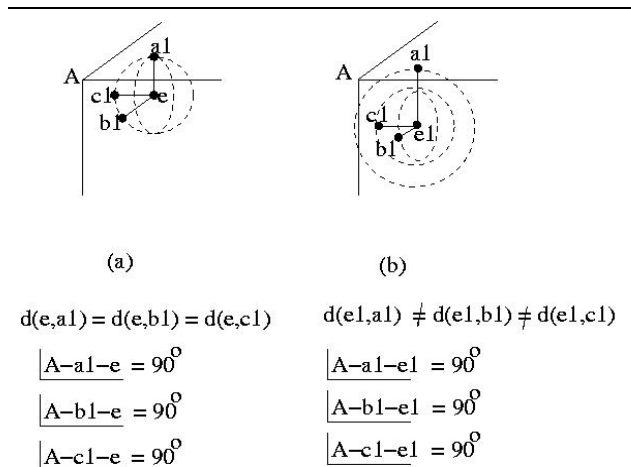


Figure 3. Distance criterion

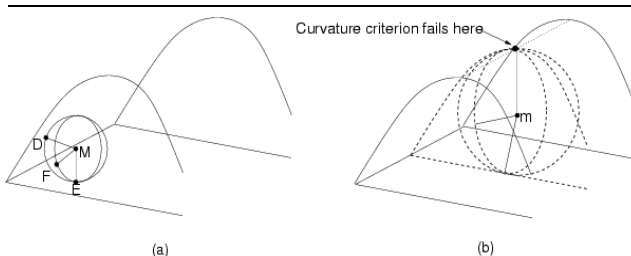


Figure 4. Curvature criterion

in Figure 3(b), even though the line segments e1-a1, e1-b1 and e1-c1 are normal to the boundary segments, they are not equal in length. So *point e1* is not a point on MAT.

Figure 4 illustrates the curvature criterion. Figure 4(a) shows that the ball is completely inside the boundary segments (i.e. does not cut the boundary segment at the touchpoint) 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 Figure 4(b) is not a MAT point but it still belongs to the set of bisector points. A point can belong to the MAT only when it satisfies both criteria.

3. Overview of the algorithm

The algorithm is similar to the one that has been developed for tracing the MAT of 2D free-form domains [18]. The algorithm constructs the MAT by tracing the seam segments. The seam segments are traced by marching along a parametric direction (referred to as the tracing parameter) of one of the faces in a triplet and solving for the touchpoints corresponding to a point on the seam. This is accomplished by solving the intersection of the normals to the surfaces in the triplet and imposing the distance criterion. Point on the MAT is obtained from the intersection of the nor-

mals to the surfaces at the touchpoints followed by a check against curvature criterion. Given that the normals are linear in contrast to the bisectors which can be rational surfaces, the computational effort and computational complexity are greatly reduced.

It is assumed that the object is such that all seam points are regular (have exactly three touch points). However, two of the touch points can lie on the same surface (implying that the maximal ball can touch more than one point on a free-form surface).

Tracing of a seam starts from a convex vertex, where all the touchpoints are known (they are the convex vertex itself). It is assumed that the input domain contains at least one convex vertex. During the tracing of seam points, at chosen intervals, the algorithm checks for the existence of a junction point. This check is a distance check with all other surfaces of the object. Distance check is done by finding the minimum distance between a point (here, seam point) and a surface. To improve the efficiency of the check for junction points, a more restricted check that gives the correct result is proposed. For determining the presence of a junction point, it can be shown that it is sufficient to check for interference with surfaces that have not been fully traced thus far. This ensures that as the generation of seam progresses, the check for junction points involves fewer surfaces.

Once the existence of a junction point is confirmed, three of the touchpoints corresponding to the junction point are known and the remaining touchpoints of the junction point are determined from the surfaces that fail the distance check. At a junction point, there will be at least four surfaces that have touchpoints corresponding to that junction point. Assuming a non-degenerate junction point (only four surfaces present at the junction point), the four triplets of surfaces are identified (note that one of the triplets is already used). Using the remaining triplets, other seam segments are traced from the junction point. The critical task here is the identification of the tracing parameter and the direction of tracing. These (parameter and direction) are fixed by determining/estimating the direction of tangent to the new seam at the junction point. Alternately, a generate and test procedure can be used with both parameters for each surface in the triplet for the new seam.

Failure of curvature criterion indicates the presence of a point with extremal principal curvature [21] in the vicinity. In such cases, the seam will terminate at the center of curvature corresponding to the point that has the extremal principal curvature. The tracing procedure used therefore originates at a convex vertex and ends at either a convex vertex or at the center of curvature corresponding to the point that has the extremal principal curvature.

The algorithm terminates when all the convex vertices in the part have been visited, no surface triplets remain to be traced at any junction point and all faces in the object have been flagged as visited.

The output of the algorithm is a set of points on each seam. Rational curves are fit to point in each seam segment. Since sheets are bound by seam segments and a sheet is generated by two faces in the object, the topology of the

sheet can be determined by grouping the seam segments where two of the faces in their governing triplet of faces are the same. The sheets can then be determined by either using bisectors of the face pair or through a numerical tracing scheme.

4. Algorithm details

The algorithm consists of the following steps.

- tracing step;
- checking for a junction point;
- tracing from a junction point;
- handling a reflex edge;
- handling a smooth edge;
- starting of the algorithm;
- termination of the algorithm;
- determination of sheets;

4.1. Tracing step

The boundary segments are parameterized such that the normal at any point on the surface is pointing outwards. Let $S_1(u, v)$, $S_2(s, t)$, $S_3(q, r)$ be the surface triplet under consideration. The tracing steps consists of the following tasks.

1. Modify the tracing parameter by the tracing step size along the direction of tracing. The seam is traced by finding the intersection of normals to the three surfaces such that the distance criterion is satisfied. The computational effort is in finding the roots of five nonlinear equations in five variables (see Appendix A). Convergence is fast as a good initial guess is always available (the previous touchpoints on the surfaces).
2. At each seam point, check if the intersection point satisfies the curvature criterion.

The intersection of normals that satisfy both the distance and curvature criterion is a point on seam. Such a point exists as the seam point is the intersection of two bisectors. Here one parameter is fixed and the other parameters are searched for such that their intersection of normals from their respective surfaces satisfy the distance criterion for point on a seam. The choice of the tracing parameter and the tracing direction is described later. The set of equations are solved using Globally convergent Newton-Raphson method (Newton-Raphson combined with Line searches and Back-tracking)

The check against the curvature criterion is a comparison of the radius function (distance between the touchpoint and the seam point) with the radius of curvature (reciprocal of maximum normal curvature) at the touchpoints. The maximum normal curvature is identified using the Gaussian (K) and Mean (H) curvatures. The procedure followed when the curvature criterion is violated at a point during tracing is discussed later during the description of tracing from a junction point.

4.2. Checking for a junction point

A seam terminates at one of the following—a convex corner, a junction point, the center of extremal principal curvature. Termination of tracing of a seam at a convex corner is straight forward as it only involves comparing the parameter values with the bounds on the same and checking if the vertex is convex or not. When the seam terminates at a junction point, however, additional checks are required to first determine the existence of the junction point and then to trace seams from the junction point. Termination at the center of curvature of a point with extremal principal curvature is indicated by either failure in curvature criterion or when the end of a surface that is not on a convex vertex (such as the touchpoint of cc in Figure 6(b)) is reached without encountering a junction point.

The existence of the junction point can be identified by finding the distance of the current seam point from the remaining boundary segments and checking if this distance is less than the radius value at the current seam point. Distance between a seam point and a surface is identified by using the minimum distance calculation between a point and a surface. The surfaces that are in the current triplet are not used for finding the junction point. Since it is extremely costly to check with all the boundary segments at all the seam points, this effort is reduced by the following. Distance check is not performed at all points generated on the seam. Checks for junction point are triggered by a different (and larger) step size called ‘junction point’ step size. It can be shown [17] that the check for junction point need not consider faces for which all the corresponding seam segments have already been identified.

Due to the different step sizes used for tracing the seam point and for checking for junction point, it is possible to overshoot a junction point during the tracing procedure. A simple bisection procedure implemented between last point on the seam where the distance check was not violated and the current point where the check is violated will identify the junction point.

4.3. Tracing from a junction point

At a junction point, there will be at least four boundary entities. Four triplets of boundary segments are possible out of the four boundary segments. Let $S_1(u_1, v_1)$, $S_2(u_2, v_2)$, $S_3(u_3, v_3)$ and $S_4(u_4, v_4)$ be the four boundary segments at a junction point. One of the triplets, say, $(S_1(u_1, v_1), S_2(u_2, v_2), S_3(u_3, v_3))$ has already been used for tracing a seam. For each of the remaining triplets the corresponding zero-set equations are solved to generate the seam. Once the triplet generates a valid seam, that triplet is marked as VISITED to avoid duplicate tracing. Note that when the triplet meets in a convex corner, there is no need to check for junction points during tracing. This is because, when the triplets meet in a convex corner, it can be shown that there will be no more junction point [17] (note however that a junction point is possible if there is a failure in curvature criterion).

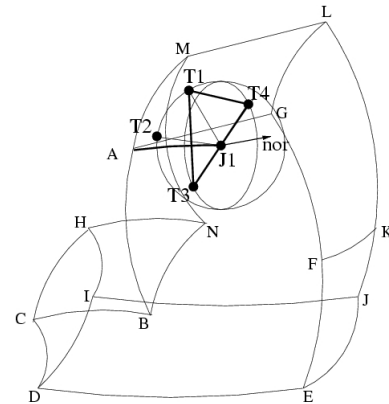


Figure 5. Identifying the tracing direction

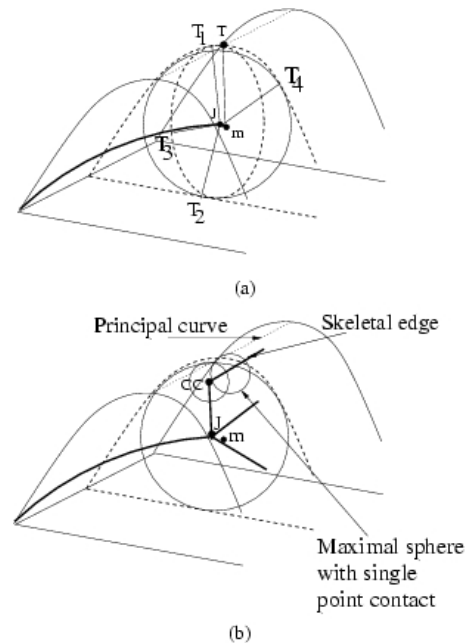


Figure 6. Failure of curvature criterion in 3D

4.3.1. Identifying the tracing direction: In this step, the tracing parameter and the direction along which tracing has to be done from a junction point is determined.

Determining the tracing direction at the junction point uses the direction of the tangent to the new seam at the junction point. The tangent to the new seam is obtained from the touchpoints on the faces in the triplet corresponding to the new seam as follows. The normal of the triangle formed by the touchpoints (refer Figure 5 where the touchpoints T1, T3 and T4 form a triangle) through the junction point is the direction of the tangent to the new seam corresponding to this triplet. The normal is projected onto the tangent plane of one of the faces at the touchpoint. This face must have also been part of the previous triplet for which seam

was traced. The tracing parameter and the direction are then determined based on the direction of the projected normal. The parameter along which the tangent is closest to the projected normal is taken as the tracing parameter. The choice of tracing direction is made based on the direction in which this parameter has moved in tracing the previous seam. The seam point can also be obtained by marching along the tangent to the seam curve[1]. This has been used in [20] but only polyhedral entities are handled. The procedure used in the present approach uses the same idea but only at junction points thereby reducing the computational effort involved in determining the tangent of the seam (controlled by six parameters).

4.3.2. Failure of curvature criterion: Failure in the curvature criterion implies that the tracing of seam has missed a junction point. Point 'm' in the Figure 6(a) is a point where failure of curvature criterion happens ('T' is the touchpoint on the surface where curvature criterion fails). The failure in curvature criterion facilitates the identification of the junction point 'J', where the fourth surface corresponding to the junction point is also one of the surfaces in the current triplet. Junction point 'J' is identified by backtracking the traced seam. During backtracking at each seam point the distance check for a junction point is performed with only the surface where the curvature criterion has failed. After 'J' is identified, the corresponding touchpoints are T_1, T_2, T_3 and T_4 , where T_4 is the touchpoint on the surface where T_1 also lies (refer 6(a)). Checking against the curvature criterion is better than including the faces in the current triplet in the check for junction point because it becomes cumbersome. From the junction point 'J', the tracing can be done as usual. However, in two of the triplets, two surfaces are same but with different touchpoints. Note that the termination of one of the seam ends at the center of curvature of the point where the curvature criterion fails. The point on the surface is called extremal maximal positive curvature point and a trace of such points on the surface gives the *principal curve*. Note that there is a seam corresponding to the principal curve called the *skeletal edge* (refer 6(b)), where the maximal spheres have a single touchpoint. Also, note that it is quite tedious to find the principal curve of any surface *a priori*. The violation of curvature criterion helps in identifying points on the principal curve.

4.4. Handling a reflex edge

A reflex edge is represented as a set of cylindrical arcs with zero radius. Associated with each arc segment is a normal. The normals are assigned to these segments in order, such that the change in the normal while crossing the reflex edge is divided uniformly across these segments. Tracing of seam points is done by suitably modifying the equations in Appendix A. The check for the presence of a junction point has to be done here as well as it is possible to encounter a junction point.

4.5. Handling a smooth edge

The domain may contain edges that are tangential, that is the edge between two surfaces is a smooth one. In such cases, the tracing procedure will not terminate when the smooth edge is reached. The tracing procedure continues with the neighborhood tangential surface and the other surfaces in the current triplet.

4.6. Starting of the tracing

Tracing starts from a convex corner. It is assumed that there is at least one convex corner present in the object. One of the parameters of any of the incident faces is chosen as the tracing parameter. The tracing direction is that which yields points inside the face.

4.7. Termination of the tracing

The termination of the algorithm occurs when all the convex vertices are visited by tracing, no triplets remain to be traced at any junction point and all the faces have been visited.

4.8. Determination of sheets

In this step, the sheets formed by the seam segments are identified using a procedure similar to that described in [20]. Loops of seams and convex edges on the object boundary are formed. Each seam will be used in more than one loop. Each loop obtained forms the boundary of a sheet. From each loop, the face pair that corresponds to the sheet is obtained by identifying the common pair of faces in the triplets corresponding to each seam in the loop. The geometry of the sheet can now be obtained by trimming the bisector of the pair of faces [9] with the bounding loop of seams. Alternately, a tracing procedure can be used with the face pair to obtain sheet points. These are then fit with a surface to the desired accuracy and trimmed with the bounding loop of seam segments.

5. Results and Discussion

The algorithm described has been implemented and this section presents the results obtained for some test objects. Currently, generation of the seams and finding the boundary of sheets only has been implemented.

MAT for some test objects having free-form boundary surfaces are shown in Figures 7 to 14 (some of the boundary surface patches have been meshed). Figure 7 shows the MAT for a human-like object. Figures 8 and 9 show the MAT for a bracket-kind of object with a concave edge in the latter. Figure 12 shows the MAT for an object having fillets. Figure 13 shows the MAT for an object having a principal curve (at its topmost part).

Robustness of this algorithm primarily depends on the tracing step size that is being chosen. It is safe to say that

no hard and fast rule exists to identify a correct step size. A small enough step size is recommended to achieve good accuracy in tracing of the seams and to avoid over stepping, but not so small that the process becomes very slow. For the objects that we have tested, we have found that a step size of 10^{-3} may be sufficient to achieve an accuracy of around 10^{-6} , though some larger step size such as 10^{-2} may be sufficient for some objects. An automatic determination of step sizes a priori looks unlikely, though a curvature based step size is certain to make the algorithm more robust.

Table 1 shows the time taken in seconds and the number of seams generated for the given objects. The implementation is done on a PIII 450 MHz with 256MB RAM. The tracing step size is 10^{-3} and the junction point step size is ten times that of the tracing step size i.e. 10^{-2} . The overall tracing time ranges in the order of few seconds (for the object shown in Figure 7) to few minutes (for the object shown in Figure 11(a)). The time scale looks very much practical (compared to the existing time scales which are quite high even for polyhedral objects) and this proves the efficiency of this algorithm. Figure 11(b) shows the MAT for the object in Figure 11(a) with a tracing step size of 10^{-4} and the junction point step size of 10^{-3} . Time taken for generating the MAT is 1105s. Though the time taken for generation of MAT is high (which is understandable as the step size is ten times smaller than the previous one), note that the topology of the medial axis is not affected by the step sizes chosen.

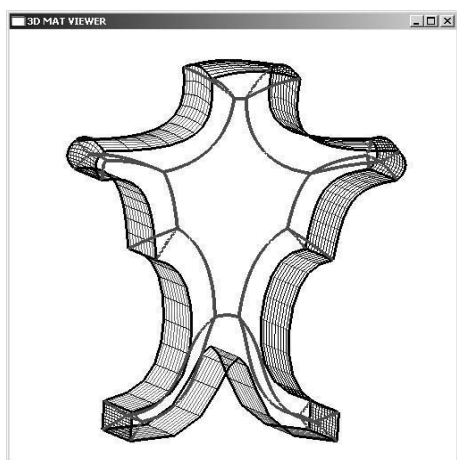


Figure 7. Test object and its 3D MAT

5.1. Discussion

5.1.1. Multiply-connected object: It has been shown that, if the faces of the solid are simply-connected, then all the sheets will be simply-connected as well [20]. When the object consists of multiply-connected faces then sheets may also be multiply connected. Earlier approaches split the multiply-connected faces into simply-connected

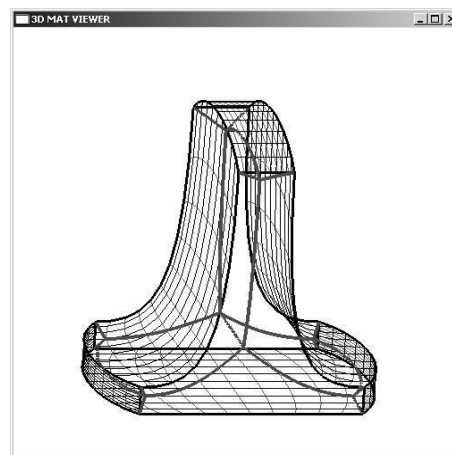


Figure 8. Test object and its 3D MAT

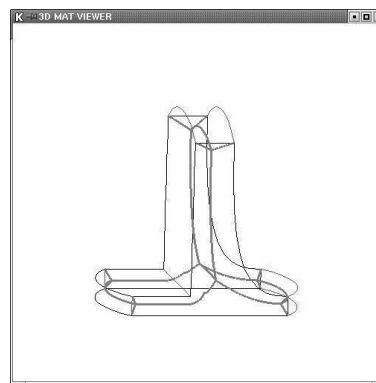


Figure 9. Test object and its 3D MAT

faces. This results in additional sheets that need to be post-processed. Moreover, splitting multiply-connected faces into simply-connected faces may be easy when dealing with planar faces but it is not the case for non-linear, free-form faces.

In the proposed approach, the faces that are handled during the tracing of seam segments are flagged. If there are convex vertices or faces that remain unflagged even though all the triplets identified at junction points have been processed, multiply connected sheets are indicated. In this case the tracing procedure is again initiated from the convex corner(s) that have not been flagged. If there are unflagged faces but no unflagged convex vertices, the present procedure cannot generate the multiply connected sheets that are indicated. MAT for a multiply-connected object is shown in Figure 14.

5.1.2. Correctness, completeness and complexity: The proposed algorithm traces the MAT of the objects by tracing the seam segment by segment. As the points on the seam are determined based on the definition of MAT, the result will always be correct. Branching from one triplet to the

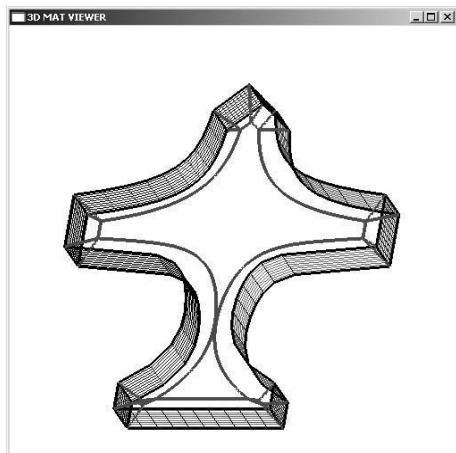


Figure 10. Test object and its 3D MAT

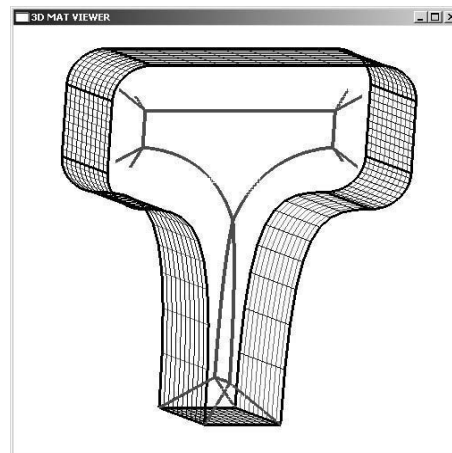
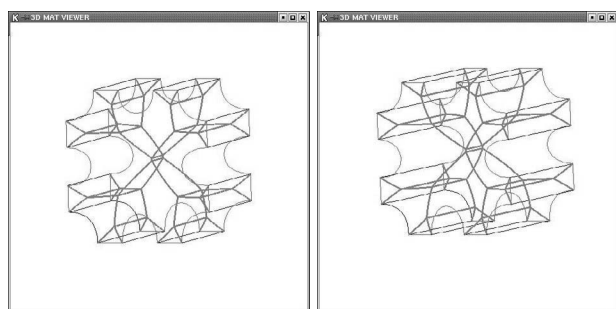


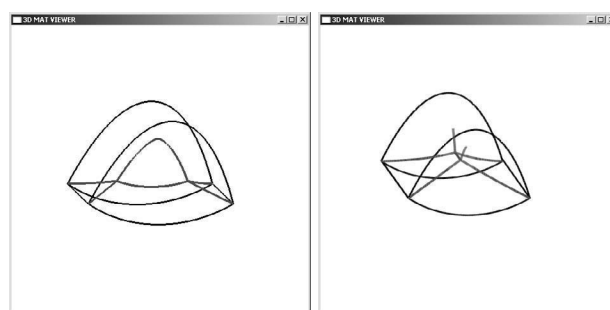
Figure 12. Test object (with fillets) and its 3D MAT (skeletal edges are not shown)



(a) Step sizes 0.001, 0.01

(b) Step sizes 0.0001, 0.001

Figure 11. MAT for the same object with different step sizes



(a) Test object and its MAT

(b) 3D MAT (skeletal edges not shown)

Figure 13. 3D MAT for objects with different thickness

other is based on the continuity of MAT.

Completeness of the algorithm is argued as follows. As the algorithm correctly identifies the junction points and each of the triplets at every junction point are traced, the algorithm generates all the seam segments when the sheets are simply connected. As discussed above, by tracking the boundary entities as well (whether they have been visited during tracing or not), the presence of multiply-connected sheets are identified correctly. The algorithm therefore does not miss any entity in the MAT and generates all the seams (and sheets) in the MAT. The overall complexity of the algorithm can be shown to be $O(n^2)$, where n be the number of boundary segments (faces and reflex edges)

Note that this procedure can be used when two or more surfaces in the triplet associated with a seam are identical. This implies that the maximal ball can touch more than one point on a free-form surface, which is crucial for objects having free-form faces. Check for the violation of the curva-

ture criterion helps in identifying the surface to be repeated in the triplet for tracing the seam. However, at present the skeletal edges are not traced. Note that the same algorithm can be used to handle objects having internal vertices by suitably modifying the equations in Appendix A.

The analysis of the algorithm with respect to the effect of numerical errors due to finite precision indicates that the algorithm is stable with respect to round-off errors. The MAT algorithm was implemented with single and double precision and there was no difference in the result to within the tolerance used ($10E - 6$). The choice of junction point step size is critical. A smaller value will ensure correctness at the cost of greater effort. A larger value may result in overshooting more than one junction point. One option could be to use a larger junction point step size and backtrack af-

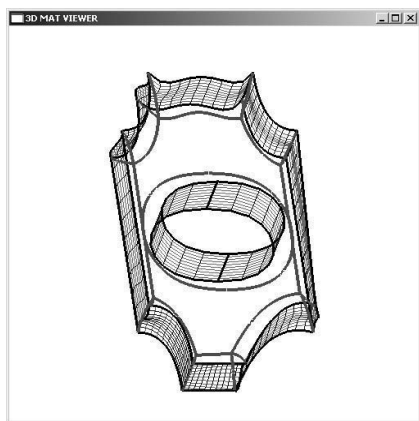


Figure 14. Test object and its 3D MAT

Figure no.	No. of seams	Time(in sec.)
8	22	36
11(a)	72	115

Table 1. Time taken for generation of 3D MAT for typical objects

ter one junction point has been identified to ensure no other junction point has been missed. If the junction point step size is chosen correctly, the algorithm is robust enough to capture an overshoot junction point. The branching decisions are therefore made correctly. In practice we have found that a value ten times the tracing step size is appropriate for the junction point step size. As long as the tracing step size is below a threshold (smallest feature in the domain), the tracing procedure will yield the correct connectivity between the MAT segments.

6. Conclusions

An algorithm for generating the Medial Axis Transform (MAT) 3D objects with free-form boundaries has been described. The algorithm generates the MAT by a tracing technique by marching along the boundary of the object. Criteria derived from the definition of the MAT are used to generate the points on the MAT and identify the junction points. The algorithm is shown to be robust and correct. Results of implementation on some test solids have been provided. Ongoing work is addressing the generation of skeletal edges and relaxing the constraint that the seam points have exactly three touch points.

References

[1] E. L. Allgower and K. Georg. *Introduction to numerical continuation methods*. SIAM, 2003.

- [2] D. Attali and A. Montanvert. Computing and simplifying 2d and 3d continuous skeletons. *Computer Vision and Image Understanding*, 67(3):261–273, September 1997.
- [3] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, pages 362–381, 1967.
- [4] T. Culver, J. Keyser, and D. Manocha. Accurate computation of the medial axis of a polyhedron. In *Proceedings of the Symposium on Solid Modeling and Applications*, pages 179 – 190, Ann Arbor, Michigan, 1999. ACM.
- [5] T. K. Dey, H. Woo, and W. Zhao. Approximate medial axis for cad models. In *Proceedings of the Symposium on Solid Modeling and Applications*, pages 280–285, University of Washington, Seattle, 2003. ACM.
- [6] T. K. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. In *Proceedings of the Symposium on Solid Modeling and Applications*, pages 356–366, Max-Planck-Institut fr Informatik, Saarbrcken, Germany, 2002. ACM.
- [7] D. Dutta and C. M. Hoffmann. On the skeleton of simple csg objects. *Journal of Mechanical Design, Transaction of ASME*, 115:87–94, March 1993.
- [8] G. Elber and M.-S. Kim. The bisector surface of rational space curves. *ACM Transaction on Graphics*, 17(1):32–39, January 1998.
- [9] G. Elber and M.-S. Kim. A computational model for nonrational bisector surfaces : curve-surface and surface-surface bisectors. In *TR-CIS9908, Center for Intelligent Systems technical reports*, Department of Computer Science, 1999. Technion-Israel Institute of Technology.
- [10] G. Elber and M.-S. Kim. Computing rational bisectors. *IEEE Computer Graphics and Applications*, pages 76–81, November/December 1999.
- [11] G. Elber and M.-S. Kim. Rational bisectors of csg primitives. In *Proceedings of the Symposium on Solid Modeling and Applications*, pages 159–166, Ann Arbor, Michigan, 1999. ACM.
- [12] M. Etzion and A. Rappoport. Computing voronoi skeletons of a 3-d polyhedron by space subdivision. *Computational Geometry, Theory and Applications*, 21:87–120, 2002.
- [13] M. Foskey, M. C. Lin, and D. Manocha. Efficient computation of a simplified medial axis. In *Proceedings of the Symposium on Solid Modeling and Applications*, pages 96–107, University of Washington, Seattle, 2003. ACM.
- [14] S. Gelston and D. Dutta. Boundary surface recovery from skeleton curves and surfaces. *Computer Aided Geometric Design*, 12:27–51, January 1995.
- [15] P. Giblin and B. B. Kimia. Formal classification of 3d medial axis points and their local geometry. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26:238–251, 2004.
- [16] D. Lavender, A. Bowyer, J. Davenport, A. Wallis, and J. Woodwark. Voronoi diagrams of set theoretic solid sodels. *IEEE Computer Graphics and Applications*, pages 69–77, September 1992.

- [17] M. Ramanathan. *Constructing medial axis transform of domains bound by free-form entities*. PhD thesis, Department of Mechanical Engineering, Indian Institute of Science, 2003.
- [18] M. Ramanathan and B. Gurumoorthy. Constructing medial axis transform of planar domains with curved boundaries. *Computer-Aided Design*, 35(7):619–632, June 2003.
- [19] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Shape description by medial surface construction. *IEEE Trans. on Visualization and Computer Graphics*, 2(1):62–72, March 1996.
- [20] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. An algorithm for medial axis transform of 3-d polyhedral solids. *IEEE Transaction on Visualization and Computer Graphics*, 2(1):44–61, 1996.
- [21] G. M. Turkiyyah, D. W. Storti, M. Ganter, H. Chen, and M. Vimawala. An accelerated triangulation method for computing skeletons of free-form solid models. *Computer-Aided Design*, 29(1):5–19, 1997.
- [22] F. E. Wolter and K. I. Friese. Local and global methods for analysis interrogation, reconstruction, modification and design of shape. In *Proceedings of Computer Graphics International Conference*, pages 137–151, Geneva, Switzerland, 2000. ACM.

A. Tracing a seam between 3 surfaces

The following is based on the derivation for bisectors in [9]. Consider the seam point B between $S_1(u, v)$, $S_2(s, t)$, $S_3(q, r)$ (denoted as S_1 , S_2 and S_3), three regular rational surfaces which are sufficiently differentiable as needed. Then B should satisfy the following equations.

$$\left\langle B - S_1, \frac{\partial S_1}{\partial u} \right\rangle = 0 \quad (\text{A.1})$$

$$\left\langle B - S_1, \frac{\partial S_1}{\partial v} \right\rangle = 0 \quad (\text{A.2})$$

$$\left\langle B - S_2, \frac{\partial S_2}{\partial s} \right\rangle = 0 \quad (\text{A.3})$$

$$\left\langle B - S_2, \frac{\partial S_2}{\partial t} \right\rangle = 0 \quad (\text{A.4})$$

$$\left\langle B - S_3, \frac{\partial S_3}{\partial q} \right\rangle = 0 \quad (\text{A.5})$$

$$\left\langle B - S_3, \frac{\partial S_3}{\partial r} \right\rangle = 0 \quad (\text{A.6})$$

$$\langle B - S_1, B - S_1 \rangle - \langle B - S_2, B - S_2 \rangle = 0 \quad (\text{A.7})$$

$$\langle B - S_2, B - S_2 \rangle - \langle B - S_3, B - S_3 \rangle = 0 \quad (\text{A.8})$$

$$\langle B - S_3, B - S_3 \rangle - \langle B - S_1, B - S_1 \rangle = 0 \quad (\text{A.9})$$

where $\langle a, b \rangle$ denote the dot product between a and b . Equations A.1 and A.2 mean that the point B is located on the normal line of S_1 . Equations A.3 and A.4 mean that the point B is located on the normal line of S_2 and equations A.5 and A.6 mean that the point B is located on the normal line of S_3 . Moreover, equations A.7, A.8, A.9 constrains the point B to satisfy the distance criterion.

Combining equations A.7 and A.9, we get,

$$2\langle B - S_1, B - S_1 \rangle = \langle B - S_2, B - S_2 \rangle + \langle B - S_3, B - S_3 \rangle \quad (\text{A.10})$$

Since B lies on the normal line of S_1 , it can be written as

$$B = S_1 + \alpha n_1 \quad (\text{A.11})$$

where $n_1 = n_1(u, v) = \frac{\partial S_1}{\partial u} * \frac{\partial S_1}{\partial v}$ is an unnormalised normal vector of S_1 and $\alpha \in \mathbb{R}$. Then equations A.1 and A.2 are automatically satisfied. By substituting for B from equation A.11 into equation A.10, we get the following equation:

$$2\langle \alpha n_1, \alpha n_1 \rangle = \langle S_1 + \alpha n_1 - S_2, S_1 + \alpha n_1 - S_2 \rangle + \langle S_1 + \alpha n_1 - S_3, S_1 + \alpha n_1 - S_3 \rangle \quad (\text{A.12})$$

$$\alpha = - \left(\frac{\langle S_1 - S_2, S_1 - S_2 \rangle + \langle S_1 - S_3, S_1 - S_3 \rangle}{2(\langle S_1 - S_2, n_1 \rangle + \langle S_1 - S_3, n_1 \rangle)} \right) \quad (\text{A.13})$$

where $\alpha = \alpha(u, v, s, t, q, r)$. Substituting the above in equation A.11 and updating the equations A.3 to A.6 and A.8, we have the following equations:

$$\left\langle f, \frac{\partial S_2}{\partial s} \right\rangle = 0 \quad (\text{A.14})$$

$$\left\langle f, \frac{\partial S_2}{\partial t} \right\rangle = 0 \quad (\text{A.15})$$

$$\left\langle g, \frac{\partial S_3}{\partial q} \right\rangle = 0 \quad (\text{A.16})$$

$$\left\langle g, \frac{\partial S_3}{\partial r} \right\rangle = 0 \quad (\text{A.17})$$

$$\langle f, f \rangle - \langle g, g \rangle = 0 \quad (\text{A.18})$$

where

$$f = f(u, v, s, t, q, r) = B - S_2 = (S_1 - S_2) - \alpha * n_1 \quad (\text{A.19})$$

$$g = g(u, v, s, t, q, r) = B - S_3 = (S_1 - S_3) - \alpha * n_1 \quad (\text{A.20})$$

The common zero-set of the equations A.14 to A.18 satisfies all the constraints of equations A.1 to A.9. Fixing one of the parameters (say ‘u’, for instance), the rest of the parameters can be identified by solving the above zero-set equations. Point on the seam is then identified by substituting the α value in equation A.11 and followed by a check against curvature criterion.