

MEDIAL AXIS TRANSFORM (MAT) OF GENERAL 2D SHAPES AND 3D POLYHEDRA FOR ENGINEERING APPLICATIONS

Y.-C. Chang, J.-H. Kao, J. M. Pinilla, J. Dong, F. B. Prinz

Department of Mechanical Engineering,

Stanford University, Stanford, CA 94305

micchang@Stanford.edu, jhkao@rpl.Stanford.edu, jmp@rpl.Stanford.edu,

jdong@Stanford.edu, fbp@cdr.Stanford.edu

Abstract

Medial Axis Transform (MAT) is a representation that encodes an object with symmetric (medial) axes in the object interior. MAT has been employed in a variety of applications such as pattern recognition of digital images, biological shape analysis and robotic motion planning. Although numerous algorithms have been proposed to determine MAT of polygonal objects, a robust model for arbitrarily shaped regions, especially suitable for engineering designs, is still an active area of research. In this paper, a 2D approach capable of efficiently constructing MAT for arbitrarily shaped 2D regions is proposed. This method can be utilized to evaluate 3D objects for a variety of applications though it does not produce a 3D MAT. Alternatively, an algorithm for calculating the MAT of 3D polyhedra is also presented.

BACKGROUND

The Medial Axis Transform (MAT) represents a bi-jjective mapping of a n -dimensional object onto a $(n-1)$ -dimensional object. The MAT labels each point on the medial axes with a metric of the dimension of the local region. One can reconstruct the n -dimensional object by sweeping a scalable n -dimensional ball along the $(n-1)$ -dimensional medial axis with its radius equal to the dimensional

metric recorded on the medial axis point. Together with the boundary representation, MAT empowers shape manipulation and geometric reasoning. The majority of the following discussion is restricted to two and three-dimensional objects only.

MAT has been employed in applications such as pattern recognition of digital images [1], finite element mesh generation [2,3], analysis of VLSI designs [4], path generation for pocket machining [5,6], medial surface extraction for engineering analysis [7], shape blending in computer animation [8], design rule checking for sheet metal components [9], punch shape recognition [10], robotic motion planning [11], feature recognition [12], biological shape analysis [13] as well as manufacturing planning and manufacturability analysis [14].

Algorithms to compute the true medial axis transform for a freeform 3D object have been studied for many years, yet robustness issues remain. One solution proposed in our paper is to evaluate a 3D object in terms of 2D slices. It is computationally economic to compute MAT of many sliced 2D regions as opposed to that of a full 3D freeform solid. Such a computation is only meaningful if the sliced 2D MAT provides adequate information for certain shape interrogations. Alternatively, it is proposed to convert a 3D freeform object into a tessellated representation and construct the 3D MAT of the resulting polyhedra. This method is computationally more expensive in comparison to the 2D approach, and for obvious reasons does not capture nonlinear features. However, it preserves the essence of the 3D object which may be of practical importance in a variety of applications.

DEFINITION OF MEDIAL AXIS TRANSFORM

The Medial Axis Transform (MAT) was first proposed by Blum [13] to describe shapes for biological problems. He defined the medial axis as loci of centers of locally maximal balls inside an object. In two dimensions, it would be the loci of centers of locally maximal disks inside the region. A ball or disk is locally maximal if there exist no other balls or disks that contain it. The points on the medial axis are called medial axis points. These medial axis points together with the radii of the associated locally maximal balls define the medial axis transform of an object.

The formal definition of medial axis transform is as follows:

Let \mathcal{A} be a subset of \mathbf{R}^n , $x \in \mathbf{R}^n$ be a point in \mathcal{A} . The medial axis transform of \mathcal{A} , $\mathbf{MAT}(\mathcal{A})$, is a subset of \mathbf{R}^{n+1} consisting of the closure of points $(x; r_x)$; $r_x \in \mathbf{R}$ such that the ball centered at x with radius r_x is locally maximal in \mathcal{A} . The set of x is the medial axis $\mathbf{MA}(\mathcal{A})$.

In other words, $MAT(A)$ is a set of $(x; r_x)$, where $x \in A$ and $r_x \in \mathbf{R}$, such that

$$1. (x; r_x) \in MAT(A) \Rightarrow Ball(x; r_x) \subseteq A$$

$$2. (x_1; r_{x1}); (x_2; r_{x2}) \in MAT(A) \Rightarrow Ball(x_1; r_{x1}) \subseteq Ball(x_2; r_{x2})$$

The balls that satisfy the above conditions are called medial axis balls.

2D-MEDIAL AXIS TRANSFORM: Clearance Function Representation

We introduce a clearance function based on the boundary representation to record the medial axis transform and proximity information. This representation directly associates boundary points to the corresponding proximity metrics so that proximity information is immediately available at any given boundary point. In addition, we found that by utilizing such a representation one can:

1. Improve the performance and simplify the implementation of MAT computation,
2. Reduce the redundancy of defining MAT when a boundary representation is also available, and
3. Provide simplified geometric operations during MAT computation.

The following paragraphs describe the proposed representation and a method to computing clearance functions and MAT.

Definition: (Clearance Function)

Let P be the set of boundary points of a connected, compact, and regular region A . The clearance function $C : P \rightarrow \mathbf{R}$ maps a boundary point $p \in P$ to the distance between p and its projection M_p . Furthermore, since the medial axis ball centered at M_p is tangent to ∂A at p ,

$$M_p = p + C_p n_p$$

, where n_p is the unit normal vector at p pointing toward the interior of A ; C_p is the distance of projection.

The clearance function maps a set of boundary points of A to the radii of the associated medial axis balls. Since each medial axis ball must touch two or more boundary points, the projection from boundaries of a given compact region completely defines the medial axis. Moreover, together with the clearance function, the projection defines the medial axis transform.

Proposition: (Medial Axis Transform and Clearance Function)

Let C be the clearance function associated with boundary of A . The medial axis $MA(A)$ and medial axis transform $MAT(A)$ can be formulated as

$$MA(A) = \{ x \mid x = p + C_p n_p, \forall p \in \partial A \}$$

$$MAT(A) = \{ (x; C_p) \mid x = p + C_p n_p, \forall p \in \partial A \}$$

,where \mathbf{n}_p is the unit normal and C_p is the clearance function value at a boundary point p .

Given the boundary representation of a compact region, the MAT can be concisely described by the clearance function associated with boundary points of the region. The task of computing MAT of a compact region is equivalent to that of computing the clearance function.

To visualize the relationship between the clearance function and MAT, we could plot the clearance functions on the boundary in the $(n+1)$ -th dimension for an object in \mathcal{R}^n . Figure 1 shows such a representation in three dimensions for a 2D object bounded by a smooth curve. The object lies on the x - y plane where the z axis denotes the clearance functions. As the z axis “bends” toward the x - y plane, the function curve becomes the medial axis.

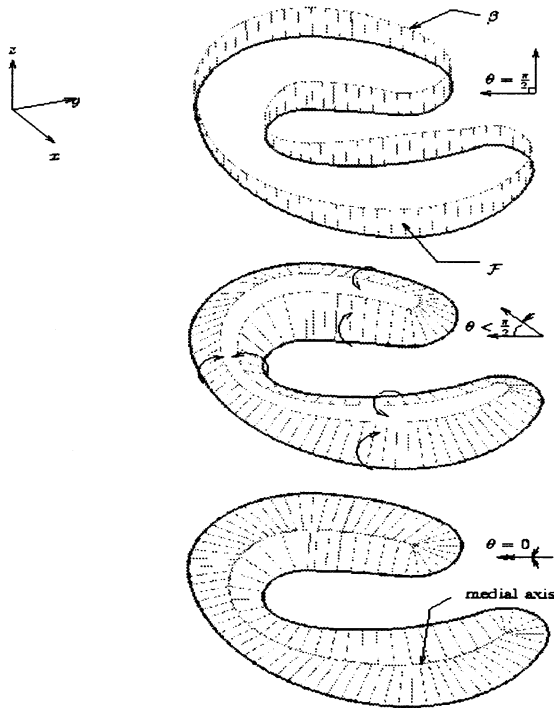


Figure 1: Visualization of the clearance function of an example 2D region. The top figure shows the clearance function attached to the boundary of the region; the middle figure shows “bending” of the clearance function in progress with the angle; the bottom figure shows that the medial axis is the result of projection of the clearance function curve.

Figure 2 illustrates the concept of clearance functions of a simple 3D object. The figure in (b) is the medial surfaces for the object depicted in (a). (c) is the boundary representation of (a) developed onto a plane, and (d) is the clearance function representation of (a).

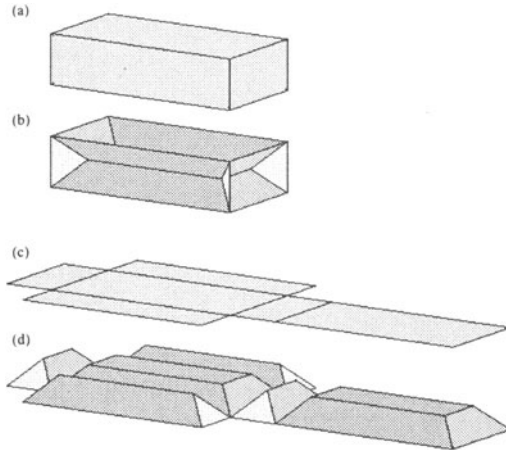


Figure 2: Clearance functions of a simple box.

It can be shown [14] that the clearance function for any given connected, compact, and regular regions exists, and is unique and continuous. Clearance functions can be computed by taking the infimum of all bisecting functions between any pair of boundary elements. A naïve approach would be to compute all possible bisecting functions and find the infimum of these functions. However, since clearance functions are continuous, one can first locate an infimum point and trace along the infimum values of the bisecting functions to construct the clearance function. For objects with holes, one could first compute the clearance function of the exterior boundary and incrementally add the interior boundary while updating all the previous clearance functions. The following figure shows the steps of computing clearance functions for an example 2D region with multiple holes.

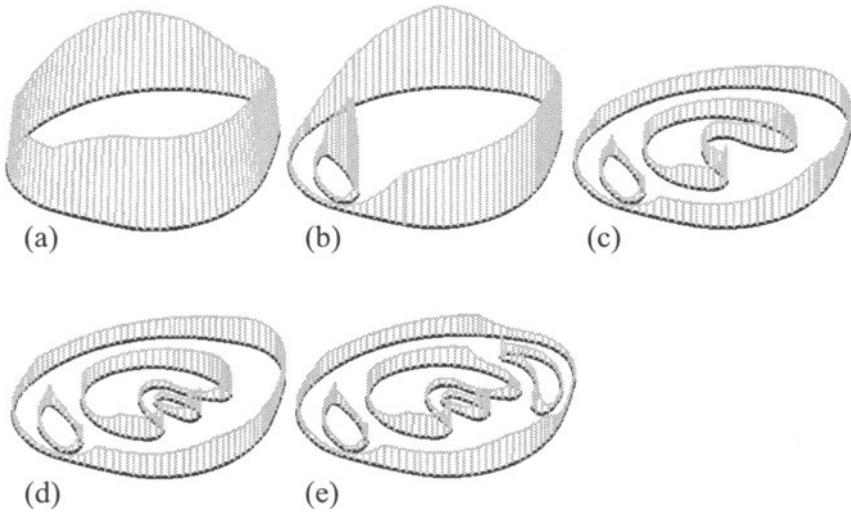


Figure 3: Clearance functions for a smooth region with holes. (a) to (e) are the intermediate results of clearance function computation by incrementally inserting an inner loop and updating the existing clearance functions.

MAT of an object is normally represented by its medial axis and the associated radius information. However, for some engineering applications the detailed geometry of medial axes is not of crucial importance. For example, one might be interested in knowing whether two features are too close to allow access to a cutting tool, and what portion of the boundary would not be machined with a given cutting tool. If one were to use the medial axis representation to locate the uncut boundary, one would need to traverse the entire medial axes, flag the portion of medial axes whose associated radii are less than the cutting tool radius, and then find the boundary elements corresponding to these medial axes. It would be convenient to have the proximity information directly recorded on the boundary so that one could directly retrieve such information by traversing the boundary.

In our work, the clearance function has been successfully applied to manufacturability analysis and automated cutting tool selection. For manufacturability analysis of roughing operations on a 3D object, one can slice the 3D geometry at the pre-determined cutting depths and generate clearance functions corresponding to exterior of the sliced cross sections. The boundary with its clearance value smaller than the radius of the rough cutting tool is therefore not accessible by the specified tool. By the same token, one can evaluate manufacturability for finish cutting operations with flat end mills. For ball end mill cutting, one would need to generate clearance functions for the 3D

model. The clearance function representation allows fast manufacturability analysis for cutting operations.

The following figure shows the model of an injection molding insert. The areas which could not be accessed are shown in black (the cutting tool is not shown in this figure).

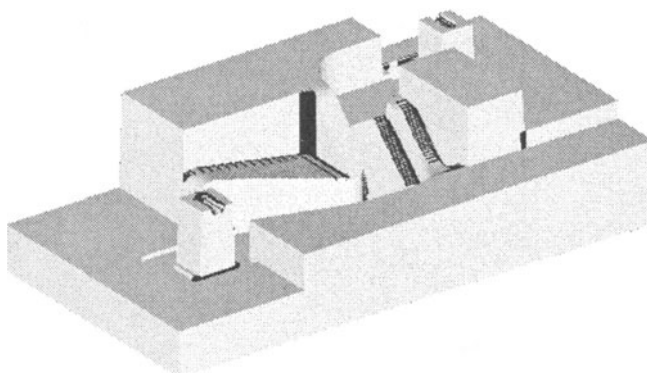


Figure 4: An injection molding insert and its manufacturability.

In addition to using clearance functions for manufacturability analysis, one can process clearance function for optimal cutting tool selection. Let us consider the case of machining time minimization. This could be formulated as a function of feed rate, tool path length, number of tool changes, and tooling change time. The most difficult part of this function evaluation is tool path length. Traditionally, machining time is computed by explicitly generating tool path for a given cutting region with a given cutting tool dimension, and then calculating the total tool path length. This approach provides more accurate tool path estimation, but may be computationally too expensive for iterative optimization cycles.

We developed a strategy of using clearance function to approximate tool path length. Given a cutting tool, one can estimate cutting area by taking the integral of clearance function within accessible cutting range. Time for machining accessible area can be approximated by dividing cutting area by tool radius and feed rate. With this approach, the clearance function is only computed once, but could be used for function evaluations during optimization iterations. The following figure shows a 2D pocket region to be machined with an optimal set of three cutters (cutters are not shown in this figure). The darkest region denotes the area to be cut by the largest cutter. The region of lighter shading is to be cut

by the second largest cutter, and the region of the lightest shading is cut by the smallest cutter.

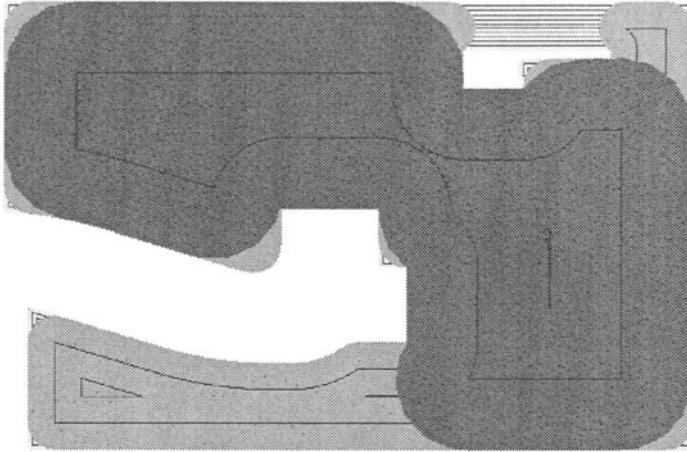


Figure 5: The results of tool selection for bulk material removal of the sample geometry.

MAT OF 3D POLYHEDRA

Influence Zone

Theoretically, the clearance function can be extended to 3D. In this case, bisectors become surfaces rather than curves. However, these surface bisectors and the medial surface may have algebraic degree greater than two, which makes the computation of 3D MAT sensitive to numerical errors. For example, as shown in Figure 6, the two bisectors which are supposed to join each other may not intersect because of some round-off errors. This tolerance problem could impede implementation of tracing algorithms which rely on accurate intersection of bisectors. We propose the concept of “influence zone” to alleviate inaccuracies of intersection calculations. In the following discussion 3D objects are represented in their tessellated forms only.

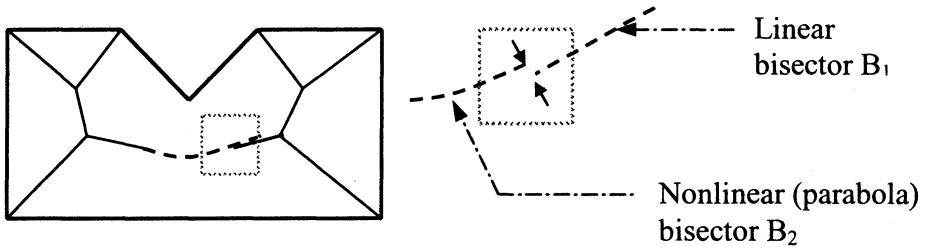


Figure 6: No intersection point is found between bisector B₁ and B₂ if some gap exists.

The “influence zone” is similar to the “influence region” of [15] and the “domain polytope” of [16] with respect to classifying the scope of “visibility” of boundary elements. The following discussion focuses primarily on how the “influence regions” interact with the object rather than how they interact with each other. In this paper the “influence zone” is defined as the region bounded by a concave boundary element, either a concave vertex or a concave edge, and a set of surfaces which are perpendicular to the adjacent boundary elements of this concave element. As it will be explained later, the medial surface section inside the influence zone is determined by this governing concave boundary element and some other boundary elements. As illustrated in Figure 7, the influence zone partitions the space in an object into three types of regions: the region inside the zone, the region outside the zone, and the region on the zone boundary. The region on the zone boundary, that is, the surface bounding the influence zone, is referred to as “influence shell”. It should be noted the present discussion does not distinguish whether each region is an open set or a closed set since it does not affect the subsequent analysis.

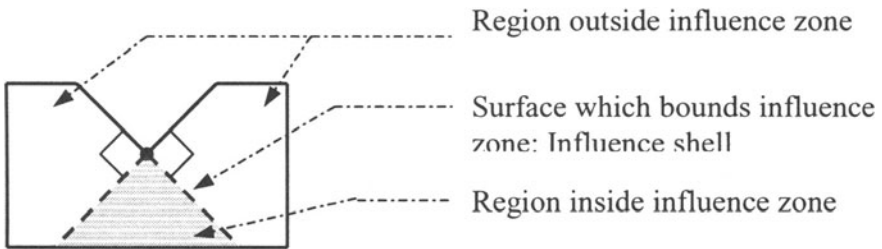


Figure 7: Influence zone partitions an object into three regions.

Since a boundary element only “influences” the shape of the medial surface within its influence region [17], any nonlinear medial surface section, if it exists, will reside inside an influence zone. This argument can be restated as follows.

Theorem 1: No nonlinear medial surface section of a polyhedron exists outside the influence zones of the object.

Proof: In a polyhedron, a nonlinear medial surface section will result only if the medial surface’s governing boundary elements are any of the following combinations: vertex-edge, vertex-face, edge-edge (if they are not on the same plane) or edge-face. Since all medial surface sections outside the influence zones are associated with face-face combination, these sections should be linear (planar surfaces in 3D).

The proposed theorem implies that the intersection between a linear bisector and a nonlinear bisector which is sensitive to numerical round-off errors can be found indirectly by calculating the intersection between the bisectors and the influence shell, as shown in Figure 8. It can be seen that the latter intersection operation will be more robust because the influence shell is always perpendicular to either bisector along the intersection.

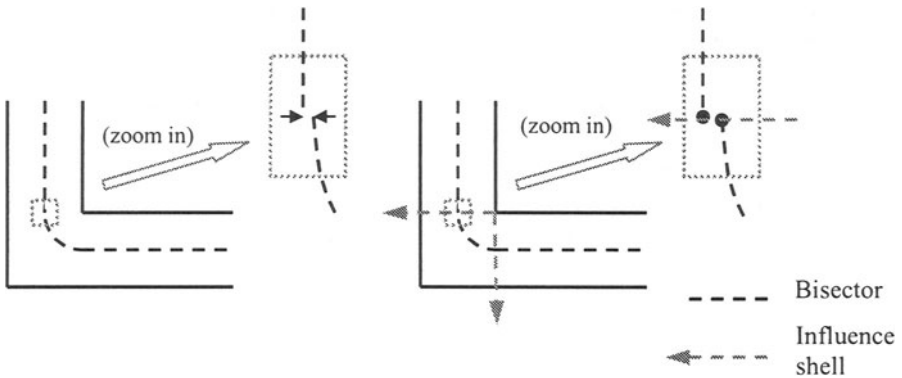


Figure 8 (a): No intersection point if some deviation exists.

(b): Intersection point exists even with deviation.

In addition, the construction of linear and nonlinear medial surface sections can be performed independently. As can be seen in Figure 9, by using the influence shell as a “bounding wall”, we can first compute the medial surface section outside the influence zone, which is done by trimming the outer linear bisectors with the “bounding wall”. Similarly, we can also trim off the inner bisectors and obtain the medial surface section inside the influence zone. By patching both sections together the complete medial surface is obtained.

Moreover, since a complete medial surface is connected, the inner medial surface section can be considered as the bridging entity which fills up the holes or gaps on the outer section. Thus for applications which weigh computation speed more than the shape precision, some approximated polygonal bridging section can be constructed instead of computing the real nonlinear inner section.

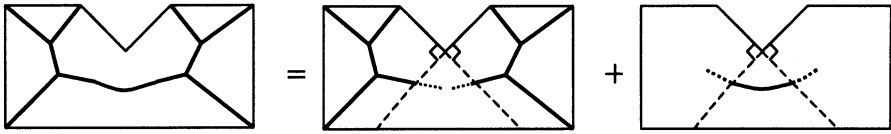


Figure 9: Linear MAT sections and nonlinear sections can be computed independently.

This influence zone can also help to reduce the complexity of computing MAT of a 3D object. The reason for this will be illustrated later. In short, the influence zone may well partition an object into more than two sub-regions. The MAT section of each sub-region outside the influence zone can be computed independently from the others. This argument is restated as follows:

Theorem 2: A polyhedron can be decomposed into a set of sub-regions by its influence zones. If there exist two sub-regions which are separated completely from each other by one influence zone, the medial surface section inside each of the two sub-regions can be computed without knowing the other.

Proof: Assume that a point P_1 on medial surface in sub-region SR_1 is governed by some point P_2 on boundary elements in sub-region SR_2 . If this statement is true, there must exist an inscribed sphere whose radius equals $|P_1P_2|$. However, $|P_1P_2|$ is larger than either $|P_1C_1|$ or $|P_1B_1|$ and thus this sphere can not be inscribed in the object. Therefore, a point on a medial surface inside SR_1 is not governed by any point on boundary elements inside SR_2 . This means the medial surface inside SR_1 can be computed as long as the boundary elements on the left side of P_2C_1 are known. The boundary elements inside SR_2 are not needed for this computation.

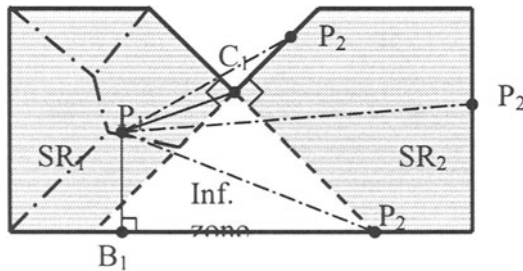


Figure 10: Left shaded region: sub-region SR_1 ; Right shaded region: sub-region SR_2 .

This theorem suggests that under certain conditions the medial surface of a polyhedron can be computed in a localized fashion, which reduces complexity of overall computation.

For example, certain 3D tracing algorithms involve a bottleneck “seam” tracing stage [18]. In that stage each “seam” (an edge on medial surfaces) is checked against all the boundary elements on the polyhedron. However, if the polyhedron can be divided into a set of sub-regions by its influence zones, each seam in one sub-region only needs to be checked against the boundary elements surrounding it. Thus the global checking is avoided and the computation cost can be reduced. It is worth noting that though the effect of the influence zone partition here is similar to the “Domain decomposition” concept in 2D domain [19] as to divide the object into smaller and more manageable pieces, our focus is to provide a simple “pre-processor” to localize the MAT construction wherever applicable. This approach can be combined with other algorithms such as the tracing scheme mentioned previously. On the other hand, the constraint of the influence zone is that it is only applicable to objects with concave boundary elements.

The steps of constructing the medial surface sections by applying the influence zone concept are illustrated briefly in the following example. As shown in Figure 11, the polyhedron is first divided into three regions: SR_1 , SR_2 , and the influence zone. We first compute the medial surface section in SR_1 utilizing the boundary elements inside both SR_1 and the influence zone. The boundary elements in SR_2 are discarded since they are irrelevant. This computation is not much different from computing the medial surface of a convex polyhedron, which is calculating the bisectors and tracing the infimum bisectors along the boundary elements in our approach. However, the bisector portion which goes outside the influence zone is trimmed off by the influence shell. The medial surface in SR_2 can be constructed in a similar fashion.

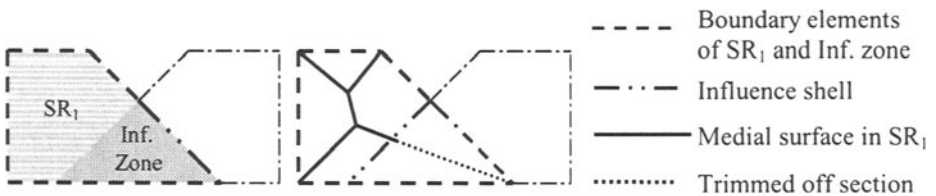


Figure 11: Medial surface in SR_1 , SR_2 , and influence zone can be computed independently.

Although the construction of the medial surface inside the influence zone seems more complicated because nonlinear medial surface portion may exist, the computation involved is still based on calculating and tracing bisectors. Similarly, we want to filter out irrelevant boundary elements in advance to localize the computation of the medial surface. Fortunately, this filtering can be done by discarding the boundary elements which are not confined by the governing boundary elements of the previously trimmed medial surface sections. As shown in Figure 12, the previously trimmed medial surface sections are MA_1 and MA_2 . Their governing boundary elements are BF_1 on one side and BF_2 and BF_3 on the other side respectively. From Figure 12 we know that the BF_1 and C_1 are the confined boundary elements. Thus we compute the bisector with boundary element BF_1 and C_1 and trim the bisector to the medial surface MA_3 .

This filtering criterion can be explained as follows. Since the mapping between medial surface and boundary elements is unique and continuous [14], the boundary elements governing a confined area A on medial surface are also confined by the boundary elements whose medial surface confines A . In our case, the inside medial surface (that is, medial surface inside the influence zone) can be considered as bounded by the trimmed medial surface sections outside, thus the boundary elements governing the inside medial surface should be also bounded by the boundary elements governing the outside trimmed medial surface. In other words, we discard the boundary element portions which have been mapped to existing outside medial surface sections when we construct the medial surface section inside the influence zone. Figure 13 shows another example where the confined boundary elements are BF_1 , BF_2 , BF_3 and C_1 .

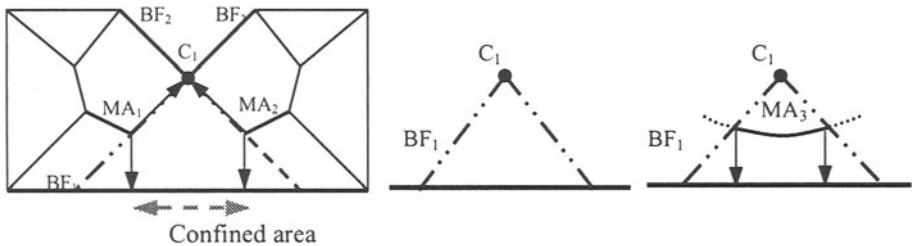


Figure 12: BF_1 and C_1 are found to be governing boundary elements of the medial surface inside the influence zone.

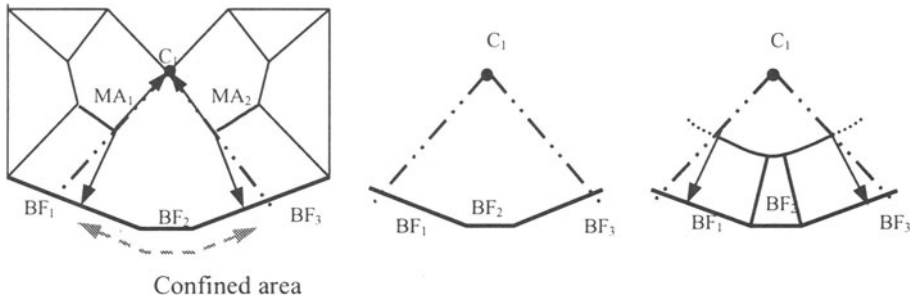


Figure 13: BF_1 , BF_2 , BF_3 and C_1 are the governing boundary elements of the medial surface inside the influence zone.

As shown in Figure 14, more complicated cases for computing medial surface inside influence zones can be found when some influence zones overlap. We can still proceed with the tracing scheme and construct the medial surface section inside the overlapping region. However, we conjecture that a more efficient computation can be achieved by characterizing the interaction among these concave boundary elements. Further study is required to understand this issue.



Figure 14: (a) Two influence zones overlap.

(b) Four influence zones overlap.

Software based on the algorithm discussed above is under development and an example result is shown in Figure 15.

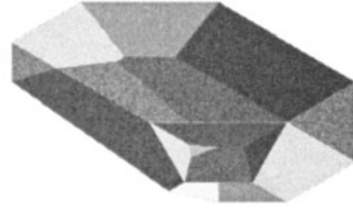
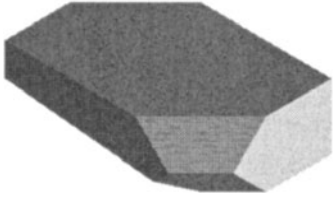


Figure 15: (a) An example polyhedron.

(b) The 3D MAT of the polyhedron.

SUMMARY

In this paper two approaches are presented to compute 2D and 3D Medial Axis Transform. The “clearance function” approach is designed to efficiently compute MAT for arbitrarily shaped 2D regions. Though this approach has not been developed to determine a ‘true’ 3D MAT, it has been utilized to evaluate 3D objects for various applications. Manufacturability analysis and automated cutting tool selection are two examples provided in this paper. The “influence zone” method is developed to compute the MAT of 3D polyhedra robustly. Though in its present form this approach is limited to polyhedra only, yet it demonstrates the possibility to compute 3D MAT in a localized fashion.

REFERENCES

- [1] U. Montanari. Continuous skeletons from digitized images. *Journal of the Association for Computing Machinery*, 16(4):534-549, October 1969.
- [2] H. N. Gursoy. Shape Interrogation by Medial Axis Transform for Automated Analysis. Ph.D. thesis, Massachusetts Institute of Technology, 1989.
- [3] N. M. Patrikalakis and H. N. Gursoy. Shape interrogation by medial axis transform. *Advances in Design Automation*, 23:77-88, 1990.
- [4] S. N. Meshkat and C. M. Sakkas. Voronoi diagram for multiply-connected polygonal domains ii: Implementation and application. *IBM Journal of Research and Development*, 31(3):373-381, May 1987.
- [5] H. Persson. Nc machining of arbitrarily shaped pockets. *Computer-Aided Design*, 10(3):169-174, May 1978.
- [6] M. Held. *On the Computational Geometry of Pocket Machining*. Springer Verlag, Berlin Heidelberg, 1991.
- [7] M. Rezayat. Midsurface abstraction from 3d solid models: General theory and applications. *Computer-Aided Design*, 28(11):905-915, 1996.

- [8] M. Shapiro and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44-50, March 1995.
- [9] R. Radhakrishnan, A. Amsalu, M. Kamran, and B. O. Nnaji. Design rule checker for sheet metal components using medial axis transformation and geometric reasoning. *Journal of Manufacturing Systems*, 15(3):179-189, 1996.
- [10] B. T. Cheok, Y. F. Zhang, and L. F. Leow. A skeleton-retrieving approach for the recognition of punch shapes. *Computers in Industry*, 32(3):249-259, March 1997.
- [11] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, Massachusetts, 1991.
- [12] R. Gadh, L. E. G^uursoz, M. A. Hall, and F. B. Prinz. Feature abstraction in a knowledge-based critique of designs. *Manufacturing Review*, 4(2):115-125, June 1991.
- [13] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, 326-380, Cambridge, MA, 1967. The M.I.T. Press.
- [14] J.-H. Kao. Process planning for additive/subtractive solid freeform fabrication using medial axis transform. Ph.D. thesis, Department of Mechanical Engineering, Stanford University, Stanford, California, 1997.
- [15] R. Joan-Arinyo, L. Perez-Vidal, and E. Gargallo-Monllau. An adaptive algorithm to compute the medial axis transform of 2D polygonal domains. In D. Roller and P. Brunet, editors, *CAD Systems Development*, 283-298, Springer Verlag, 1997.
- [16] T. Culver. Computing the medial axis of a polyhedron reliably and efficiently. Ph.D. thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 2000.
- [17] R. Joan-Arinyo, L. Perez-Vidal, and J. Vilaplana. Computing the Medial Axis Transform of Polygonal Domains by Tracing Paths. *Llenguatges i Sistemes Informàtics Report LSI-99-8-R*, Universitat Politècnica de Catalunya, April 1999.
- [18] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):44-61, March 1996.
- [19] H. I. Choi, S. W. Choi, and H. P. Moon. New algorithm for medial axis transform of plane domain. *Graphical Models and Image Processing*, 59(6):463-483, November 1997.