

# Skeletons for Representation and Reasoning in Engineering Applications

W. R. Quadros<sup>1</sup>, B. Gurumoorthy<sup>1</sup>, K. Ramaswami<sup>2</sup> and F. B. Prinz<sup>2</sup>

<sup>1</sup>Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India; <sup>2</sup>Department of Mechanical Engineering, Stanford University, CA, USA

**Abstract.** *Skeletal transforms (of which the Medial Axis Transform (MAT) is the most popular) have been proposed in the literature as a useful shape abstraction tool. This paper develops this proposition with a description of the areas where skeletons (and the MAT in particular) could serve as an enabling representation of part geometry. The paper then addresses two issues that need to be resolved for the application of skeletons in the area of engineering analysis, numerical simulation-based analysis of design in particular. These are the generation of skeletal transforms for complex 2D and 3D geometries, and quadrilateral mesh generation. Generation of the midsurface abstraction of 3D objects (which is a skeleton with MAT like properties) and MAT for 2D closed regions is then described. The paper then presents an algorithm for 2D quadrilateral mesh generation based on the MAT. The paper concludes with a discussion of the outstanding issues in the use of skeletons/MAT in design and engineering applications.*

**Keywords.** Digital thinning; Medial axis transform; Mesh generation; Mid-surface; Punching; Skeletal transform

---

## 1. Introduction

Most of the bottlenecks in any design or engineering task occur in the transition from one sub-task to another. In design, the transition from model construction to analysis has been found to be taking more time than the analysis itself. This is primarily because the information in the model has to be converted to a description of part geometry (*mesh*) that can be used by the programs performing the analysis. This task requires human intervention for successful transition. Similarly, in manufacturing planning the main problem appears to be the conver-

sion of part data to a form that can be used directly by any planning software.

Abstraction of shape information therefore becomes an important requirement for any automated or computer assisted environment for design and manufacturing. The level of abstraction required varies depending on the task for which the shape information is required. A CAD model that is typically output by all CAD systems does not explicitly contain and does not enable automatic extraction of shape information required by application domains. There are, however, representations able to facilitate applications such as manufacturability analysis, FEM mesh generation and process planning. This paper focuses on one such representation, generally referred to as skeleton transform. The Medial Axis Transform (MAT) is the most popular among such skeleton transforms. The Medial Axis Transform was first introduced by Blum [1] to describe biological shape. In simplified terms, the MAT maps outside surfaces of an object onto another surface placed halfway between the outside surfaces. The distance established between object and skeleton is part of the mapping scheme. The CAD model and the MAT are bijective, in other words, the original CAD model can be reconstructed given the MAT. A more detailed explanation of the MAT and its properties will be given in the following section.

## 2. Medial Axis Transform (MAT)

The MAT is described through a medial axis and a radius function. The Medial Axis (MA), or skeleton of the set  $D$ , denoted  $M(D)$ , is the locus of points inside  $D$  which lies at the centers of all closed discs (balls) which are maximal in  $D$ , together with the limit points of this locus. A closed disc (or ball in 3D) is said to be *maximal* in a subset  $D$  of the 2D (or 3D) space if it is contained in  $D$ , but is not a

---

Correspondence and offprint requests to: F. Prinz, Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA. Email: fbp@cdr.stanford.edu

proper subset of any other disc (ball) contained in  $D$ . The radius function of the medial axis 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 Medial Axis Transform (MAT) of  $D$  is the MA together with its associated radius function.

The boundary and the corresponding MA 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 MA segments will be one of the conic sections [2]. 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 two-dimensional MAT defines a unique, coordinate-system-independent decomposition of a planar shape into lines, and the three-dimensional MAT simplifies a solid model into a collection of surface patches. The Medial Axis (MA) of a figure is therefore also called the *skeleton* or *symmetric axis* of a part/shape.

## 2.1. Properties of MAT

- *Uniqueness*: there is a unique MA for a given object, and it is invariant with respect to the coordinate system.
- *Decomposition (Dimensional reduction)*: the dimensionality of a MA is lower than that of its object. 2D MAT transforms planar figures into lines and 3D MAT decomposes an object into surface patches.
- *Homotopical equivalence*: the MA is homotopically equivalent to the object. That is, the number of holes, enclosed voids, etc., remain the same.
- *Recomposition (Invertibility)*: with the axis and its radius function, one can reconstruct the figure by taking the union of all circles centred on the

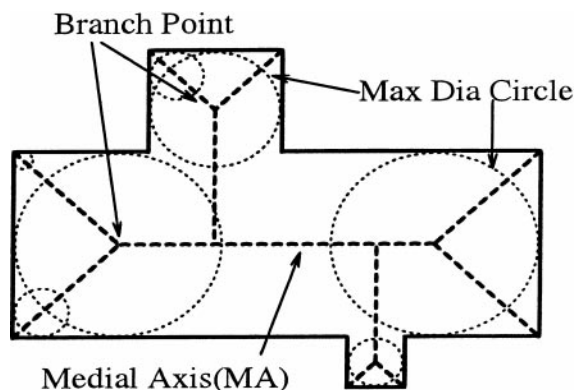


Fig. 1. Boundary and its Medial Axis (MA).

points corresponding to the axis, each with a radius given by radius function.

- The MAT has no interior.

## 3. Applications of Skeleton/MAT

The more general term of skeleton is used along with the usage of the term MAT primarily because of the intractability of the problem of generating MAT for 3D objects. While this is still a subject of research, variants of the MAT are being used as skeletons in some restricted domains, such as thin walled plastic components [3]. The term MAT will be used in this section and the remainder of the paper as the preferred skeleton. Variants when used will be referred to explicitly.

The original application for which MAT was proposed was biological shape measurement [1]. Since then MAT has been proposed and used for design/evaluation of moulds and dies, finite element mesh generation [4,5], feature extraction and path planning.

### 3.1. Mould and Die Design

The design of plastic parts (manufactured by injection moulding) accounts for almost half of today's mechanical CAD activities. Parts designed for injection moulding are typically thin walled, hence numerical simulation of the mould filling process takes advantage of the plate-like geometry. A plate-like shape may be approximated by 'collapsing' a 3D CAD model to a collection of surfaces, thus reducing the dimensionality of the problem. The collapsed surfaces together with the thickness of the thin walled part can be used to emulate the expanding mould flow front. This approach has been adopted by popular FEM packages such as Mold flow and Cadmould which assist designers in determining the feasibility of a given geometry for injection moulding. These software packages are able to predict the mould filling pattern, heat flow, warpage and distortion analysis. The preparation of the input files (referred to as analysis-model generation in the trade [3]) requires considerable skill and time from a human user, thus becoming a bottleneck in deeper penetration of numerical simulation in design. The MAT can significantly facilitate automatic generation of the mid-surfaces and mesh on these mid-surfaces for injection moulding.

### 3.2. Hexahedral Mesh Generation

Skeletons can serve as a powerful tool for FEM mesh generation. Tetrahedral mesh generation algorithms for 3D geometries have been available to the FEM community for some time. However, hexahedral mesh generators which are more desirable for numerical analysis are still under development. We envision developing an algorithm which would first determine a quadrilateral mesh on the surface of the MAT (MAT of a 3D object being primarily a collection of surfaces) [6]. The 3D mesh will be created by an advancing front technique starting from the inside, propagating outward to the model surfaces. Robust skeleton schemes are anticipated to enable the creation of a new class of hexahedral mesh generation algorithms for a broad user base [7].

### 3.3. Feature Recognition

Automatic recognition of form features in CAD representations for heuristic manufacturability analysis is another area where skeletons can serve a crucial role [8]. Features are a formal representation of patterns in form or in function that can be associated with domain knowledge. In three-dimensional mechanical design, features might include holes, fillets, slots, keyways, bosses and pockets. Identifying features from a geometric model is important in automating process-based reasoning. While feature extraction directly from geometry has been studied extensively [9] difficulties still remain in the areas of interacting features and features across different domains. If an object were to be expressed in terms of its skeletal model, the task of recognition has been shown to be more efficient [10]. Features could be discerned from patterns in the skeletal model. The efficiency of this approach is due to the fact that the patterns have to be defined in 2D as opposed to the direct feature extraction route. Figure 2 shows a rib sitting on a base and the

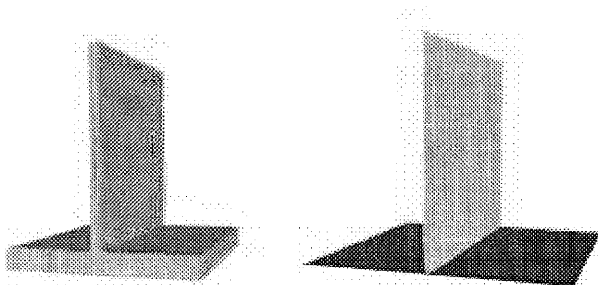


Fig. 2. A 2D part model and its skeleton.

corresponding skeletal model. In the dimensionally reduced skeletal model, the rib feature can be easily recognised by searching for neighbourhood information only.

### 3.4. Motion Planning for Material Deposition in Layered Manufacturing and Machining

In layered manufacturing, materials are deposited layer by layer along the build direction. In order to produce isotropic material deposition, contour parallel patterns are usually preferred over direction parallel patterns. However, quality of deposition paths is often constrained by its cross sectional geometry. Recently, a new class of layered manufacturing has been developed that involves both material deposition and machining and need not deposit materials in its net shape. This allows layer cross sections to be relaxed or augmented to accommodate the quality of deposition paths. An algorithm based on the radius function associated with the skeleton can be used to produce such relaxed geometry and to compute the deposition paths simultaneously [11].

Another area of application for MAT is machining where it is required to determine a cutter or combination of tools necessary to remove all excess materials in a region. A small cutter may be needed, especially near the boundaries with high curvature. Meanwhile a large cutter may be preferred to improve the machining efficiency. However, to identify these parameters, machining paths often need to be recomputed for every parameter evaluation. Given the MAT of the region to be machined, these parameters can be calculated in one pass. The radius function at a point on the MA serves as a measure of the clearance of the point to the contour. A frequency spectrum of the radius function at points along the MA provides a range of cutter sizes that can be used to machine varying portions of the region to be machined.

In addition to the applications described above, the skeletal model also has numerous applications in such diverse areas as computer vision [12], robot path planning [13], machine cutting, wire layout, computer graphics and medical diagnostics.

In this paper, we illustrate the use of skeletons in enabling numerical simulation of a manufacturing process for evaluation a design. The main problem being addressed is the automatic generation of analysis-model. A midsurface abstraction (that has the required properties of the MAT) is proposed as the skeleton for the 3D object. The MAT of the mid-surface (which could be composed of several

patches) is then used to generate an all quadrilateral mesh that is used for analysis task such as simulation of formative processes such as casting/injection moulding.

In the following section we describe algorithms to obtain the 2D MAT and midsurface abstraction as the skeletons for general closed 2D regions and 3D geometries, respectively. We then present a new algorithm to generate an all quad mesh on a set of surface patches. We present results of implementation, and conclude by identifying open issues in the use of MAT for representing and reasoning about geometry and shape.

#### 4. Construction of Skeleton – Midsurface

In this section we describe an algorithm to construct the skeleton of a closed region in 2D/3D domain. We believe that this algorithm is better suited for extension to 3D when compared to the techniques reported in the literature.

Note that many MAT algorithms attempt to compute Voronoi Diagrams (VD) first. In a subsequent step the MAT is extracted from VD. For simple polygons (without holes), Lee [14] proposed a divide-and-conquer approach to compute the Voronoi diagrams. Persson [15] and Held [16] implemented the wavefront-propagation approach for simple curvilinear polygons whose boundaries consist of linear and are segments. All of these methods first constructed Voronoi diagrams of subsets of the original regions and incrementally merged them.

For multiply-connected polygons (with holes), Held [16] and Srinivasan and Nackman [17] incrementally merged the Voronoi diagrams of individual contours to construct the complete Voronoi diagram. Gursoy and Patrikalakis [4] directly computed the MAT primarily through the identification of branch points.

Robust algorithms for 3D MAT that can handle industrial strength parts are still subject of research. In the following we describe a new approach called *digital thinning* that generates a skeletal representation of a 3D object. The algorithm approximates a 3D object by cuboidal elements from which a robust skeleton, referred to as the midsurface is derived. The midsurface is a subset of the MAT with all branching surfaces, that intersect the boundary of the original 3D object, and higher order entities removed. Higher order entities for an object represented by cuboidal elements will occur at reflex corners. The absence of such entities in the midsurface presents no problems as these entities are

removed for most reasoning involving the MAT. This midsurface has the properties of the MAT that make it useful for engineering design (namely, homotopic equivalence, dimensional reduction and invertibility).

Given an object and a norm (the box norm) [18], the closure of the set of centers of maximal cells contained in the object is defined to be the skeleton/midsurface of the object, denoted by  $S$ . Associated with this skeleton is a radius function  $r$ . Analogous to the radius function associated with the MAT, the radius function of the skeleton gives the size of the maximal box (cell) at each point in the skeleton.

The digital thinning algorithm uses a discretization of the object into cells (pixels in 2D and voxels in 3D). A quadtree (octree in 3D) algorithm can be used to subdivide the model into squares/cubes of varying size. Then the larger squares/cubes have to be subdivided such that all the squares/cubes are of the same size. The final set of squares (pixels) or cubes (voxels) are stored in the form of a graph (along with the corresponding face, edge and vertex information as applicable). In the implementation, a ray tracing technique has been used to directly subdivide the model into voxels.

For any 2- or 3-dimensional graph, we can define an element-dual for each element of a graph. For two-dimensional graphs, this is the dual frequently referred to in graph theory. For our purposes, we define a dual for each element of a 2D graph as follows:

- The dual of a pixel is a point located at its centre.
- The dual of an edge is another edge at right angles to it. The endpoints of the dual edge are the element duals of the two pixels adjacent to the original edge.
- The dual of a vertex is a pixel, bounded by the element-duals of edges incident at that vertex.

In three dimensions, the dual of a voxel is a vertex at its centre, the dual of a face is an edge passing through its centre and perpendicular to it, and so forth.

Since the vertices of a graph lie on a regular grid (of size 1) spanning space, it is immediately obvious that the set of points at the centres of pixels (2D) or voxels (3D) is another such grid, each vertex of which has coordinates which differ by 0.5 from those of the nearest vertices of the original graph. This implies that the set of element-duals of all the elements of a graph is itself a graph. This is illustrated in Fig. 3. The frame on the left shows the thinning steps. The frame on the right shows the

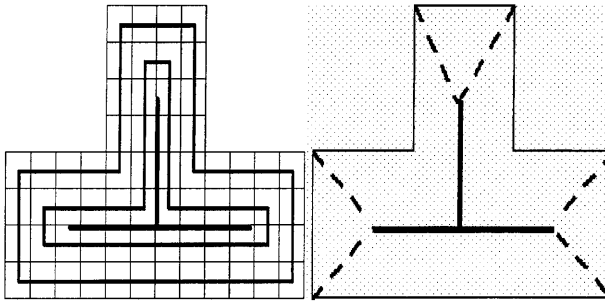


Fig. 3. Digital thinning algorithm.

final midsurface obtained. The dotted lines in this frame correspond to the branching segments present in the MAT of the object shown.

Furthermore, since the grid on which the vertices of a graph lie has a unit spacing, it is clear that displacing a graph along any axis by unity yields the same graph, and hence that the dual of the dual of a graph is the same as the original graph. This is significant from the perspective of reconstructing the original geometry (discretization) from the skeleton. In effect, the same procedure (construction of a dual of a graph) could be used to construct the object given its skeleton.

The key idea in the method of thinning employed here is that the union of the element-duals of the interior elements of a discrete object is always a thinner version of the object. The skeleton or the midsurface is obtained through successive application of the thinning process to the result of the previous step until the interior of the graph is empty while ensuring proper connectivity. The thinning process is simply the computation of the dual of the current graph. The midsurface abstraction is represented as set of connected points in 2D and a mesh of triangles in 3D, respectively. Figure 4 shows the solid model, the midsurface abstraction obtained by the digital thinning algorithm. Depending on the choice of the resolution, this part takes of the order of ten minutes on a HP715

workstation. The strengths of the digital thinning algorithm originate from the robustness which numerical digitisation offers. On the other hand, the nature of numerical digitisation creates surface steps on the midsurface that are undesirable for numerical analysis. Hence, we smoothen the surface steps by repositioning the vertices based on mesh optimization algorithm suggested by Hoppe *et al.* [19]. This algorithm has limitations due to the occurrence of local minima which may lead to incorrect results and long computational times. Figure 4(c) shows the mesh obtained after smoothening the midsurface obtained by the thinning algorithm in Fig. 4(b).

Another inherent weakness in the thinning algorithm is that the resolution of the grid has to be set based on the smallest feature size in the model. Automatic estimation of the smallest feature size *a priori* has not been successful, and therefore user intervention is required to start the algorithm. It must be mentioned that this is a crucial input, as over-estimating the resolution may require excess computational resources while under-estimation may ignore shape features smaller than the specified resolution.

## 5. Mesh Generation using MAT

In this section, we describe the application of MAT to generate meshes that can be used for finite element analysis. The presentation will focus on quadrilateral mesh generation, and conclude with a discussion on the extension of the algorithms to address hexahedral mesh generation. Simplicial three-noded triangular elements have linear shape function, and the displacements are linear, resulting in constant strain (and stress). Quadrilateral elements, on the other hand, have bilinear shape function, and therefore give a better result.

From the literature [20], unstructured quadrilateral meshing algorithms can be grouped into two main

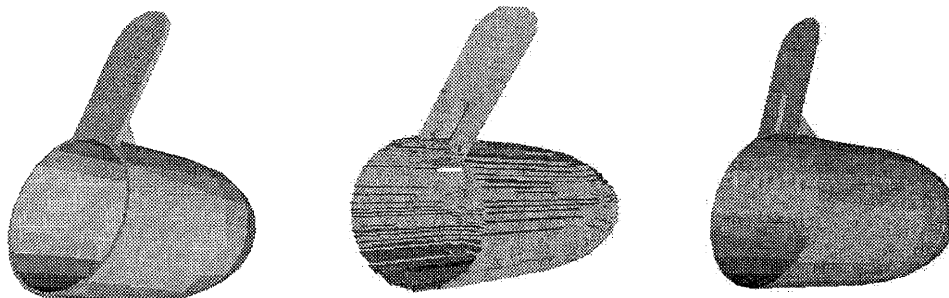


Fig. 4. Midsurface abstraction obtained by digital thinning algorithm.

categories: direct and indirect approaches. In the indirect approach, the domain is first meshed with triangles. Various algorithms are then employed to convert the triangles into quadrilaterals.

In the direct approach, quadrilaterals are placed on the surface directly using techniques based on decomposition, MAT and advancing front of elements. Talbert [21] introduces the decomposition technique: with this approach, the domain is recursively subdivided into simple polygonal shapes. The resulting polygons satisfy a limited number of templates into which quadrilateral elements are inserted. Chae [22] has recently proposed enhancements to Talbert's algorithm, with similar work presented by Nowotny [23].

Joe [24] also utilises decomposition algorithms to decompose the area into convex polygons. Using techniques previously developed for triangle mesh generation [25], Joe constructs a boundary constrained quadrilateral mesh within each convex sub-domain of the area.

Quadrilateral meshing utilising a medial axis decomposition of the domain was first introduced by Tam [26]. Having decomposed the area into simpler regions, sets of templates are then employed to insert quadrilaterals into the domain. Linear programming techniques are used to maintain compatibility of element divisions between adjoining regions of the domain.

The paving algorithm introduced by Blacker and Stephenson [27] generates complete rows of elements starting from the boundary and working in. Methods for projection of nodes, handling of special geometric situations and intersection of opposing fronts are discussed. Cass [28] further developed paving by generalizing the method for three-dimensional surfaces.

There are a number of quad meshing systems commercially available (either stand-alone or as part of an analysis package) [20]. A majority of them are based on a mapped meshing approach that requires either human intervention or some feature recognition to ensure that opposite edges have equal divisions. Quad meshing systems based on the advancing front technique require mixed element types at the junctions, and is quite time intensive. The indirect quad meshing schemes have many irregular nodes, and the resulting mesh may not be aligned with the boundary.

We propose a new algorithm, called the Punching Algorithm, that belongs to the category of direct methods, and combines the merits of both the above-mentioned direct approaches. The objective of this algorithm is to generate an all quad mesh which

contains more elements closer to a square, on a given set of connected planar patches. The mesh generated would have the characteristics of a good mesh, as proposed by Blacker [27] and reproduced below:

- *Boundary sensitive*: mesh contours should closely follow the contours of the boundary. This characteristic is of particular importance, since well-shaped elements are usually desirable near the boundary.
- *Orientation insensitive*: rotating or translating a given geometry should not change the resulting mesh topology. A mesh generated in a transformed geometry should be equivalent to the original mesh transformed.
- *Few irregular nodes*: this is a critical mesh topology feature, because the number of elements sharing a node controls the final shape of the elements, even after smoothing. Thus, a mesh with few irregular nodes, especially near the boundary where element shape is critical, is often preferred.

Robinson [29] measured the quality of elements using convexity check, skew angle, aspect ratio, warp angle and taper. For quadrilateral elements, a square is found to be the best element. In the case of a hexahedral element, a cube is the ideal element which has all normalised factors as 0.

The algorithm described in the following sections is robust enough to handle concave polygons, multiply connected patches, non-manifold planar set of patches, etc. The punching algorithm can also be used to generate adaptive quad mesh using the radius function available with the MAT [6].

## 6. Punching Algorithm – An Overview

The punching algorithm exploits the two-way mapping between the entities constituting the MAT (edges and vertices in 2D) and the boundary of the object (edges and vertices for planar 2D objects). The discretization of one boundary edge is realized first. A *strip* is then laid between a segment on the boundary and its corresponding points on the MAT. The strip then bends at the MAT towards the other boundary segment corresponding to the MAT (see Fig. 5). The laying of such a strip continues to move between a patch boundary and the MAT segment, and then between the MAT segment and a patch boundary. The laying of a single strip terminates when the strip returns to the original starting edge segment or when it encounters a

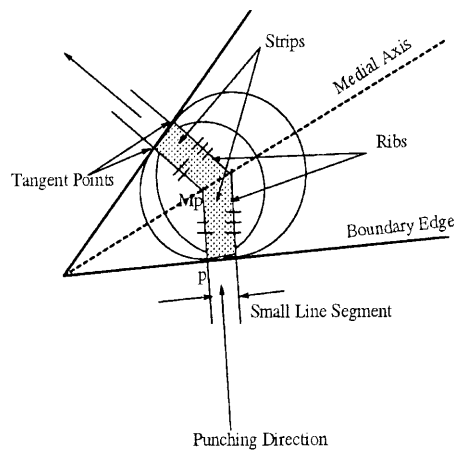


Fig. 5. Punching of line segment from the boundary.

boundary segment that has no adjoining patch or at the concave vertex. The process continues till the entire region (corresponding to the planar patches) is covered by similar strips. This process of laying a strip is referred to as a *punching operation*. Each strip can now be divided into a number of quadrilaterals, depending on the size by just marching along the edges of the strip. Merging of a pair of opposite last non-quad elements meeting at the medial axis segment, and merging two triangles in the case of concave corners, results in a quad mesh. It is very rare that any mesh generation algorithm will generate an optimal mesh without post-processing. In this case, we need smoothing only at the medial axis zone for some cases. Refinement can be done to get the final adaptive mesh.

This approach is similar to the advancing front method, and results in good quality elements at the boundary, whose contours in general follow the boundary of the domain. The advantage with this approach is that it eliminates the expensive interference checks required in the advancing front method.

Figure 5 shows two maximal diameter circles whose centre lies on the medial axis and touching the boundaries at tangent points. We call the lines joining the centre of the circle and the tangent point ribs. A line segment on the boundary is punched perpendicularly from the boundary till it meets the medial axis; it then takes a turn and comes out of the perpendicular from the other end. In the case of 2D space, punching a small line segment from the boundary edge results in a strip, whereas in the case of 3D space, punching a quadrilateral element from the boundary face results in a column.

## 7. Punching Algorithm – Details

The input to the algorithm is a set of planar patches along with the MAT of each patch. The main steps in the algorithm are:

- construct the two-way mapping between the patch boundary and the MAT,
- lay the strips starting from a boundary edge (*source-node*),
- generate the quad mesh,
- adjust the all quad mesh obtained above to generate an adaptive quad mesh.

Details on each step follow.

The input set of planar patches is first processed to obtain connectivity information between the patches. This information is required for laying the strips across patches. Each patch is treated as a node in a dual graph, and a traversal along its outer edges finds the nodes (patches) connected to it.

The MAT of each patch is used to construct the two-way mapping between the boundary edges and the medial axis segments. The mapping essentially constructs the two-way correspondence between the MAT segments (MA-SEG) and the boundary segments (MAP-SEG). End points of each mat segment are projected onto the patch boundary to identify the corresponding boundary segments. Concave vertices (internal angle  $\geq \pi$ ) are represented as dummy edge segments to record the mapping of the corresponding MAT segments (Fig. 6).

The mapping between the patch boundary and the MAT of the patch is shown in Fig. 7.

Once the two-way mapping is available, it is possible to lay strips covering the patches. The width of each strip controls the width of the quad element finally obtained. Initially, the width is set

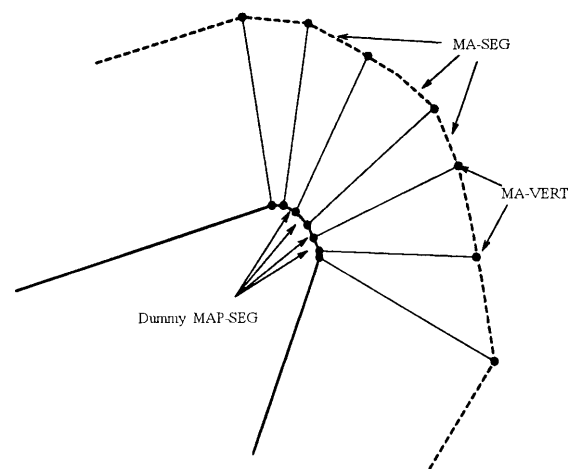


Fig. 6. Dummy MAP-SEG at concave VERT.

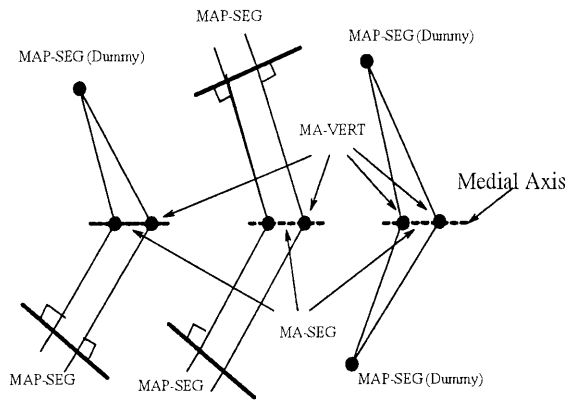


Fig. 7. Two-way map between MA-SEG and MAP-SEG.

to be some reasonable value, to be adjusted later in this step, and again during the adaptive meshing step. Portions of the boundary segments corresponding to each MAT segment (MAP-SEG) are divided into a number of segments, roughly of width  $w$ . Dividing the boundary segments corresponding to each MAT segment (as opposed to an entire boundary segment of a patch) ensures that the strips are laid at the branch points in the MAT. This is of help in the merging of non-quad elements near the MA. As the length of the edges of MAP-SEG need not be integral multiples of the chosen  $w$ , the number of divisions are adjusted to achieve the above objective. Therefore, the width of the strip is only roughly of width  $w$ . In the following description, the divisions in the segment MAP-SEG are referred to as *CUT-VERT*, and the portion of the MAP-SEG between two CUT-VERTs is referred to as *CUT-SEG*.

The punching process (to lay a strip) now requires the determination of the points on the MAT segments corresponding to each CUT-VERT. These points are referred to as *MA-CUT-VERT*. A strip is now formed by lines joining successive CUT-VERTs and their matching MA-CUT-VERTs (these lines will be referred to as *ribs* hereafter), CUT-SEG and the portion of the corresponding mat segment between the two MA-CUT-VERT (referred to as *MA-CUT-SEG*), taken in order. The punching process can now continue from the MAT segment to the other boundary segment corresponding to it, and further propagate to other patches. The punching (strip laying) process terminates when the original segment CUT-SEG is encountered (Fig. 8), or when there are no other patch incidents at a boundary edge. A proof for this is available elsewhere [6].

Once the punching process has encountered each division in the boundary of the patches, the entire region is guaranteed to be covered with strips.

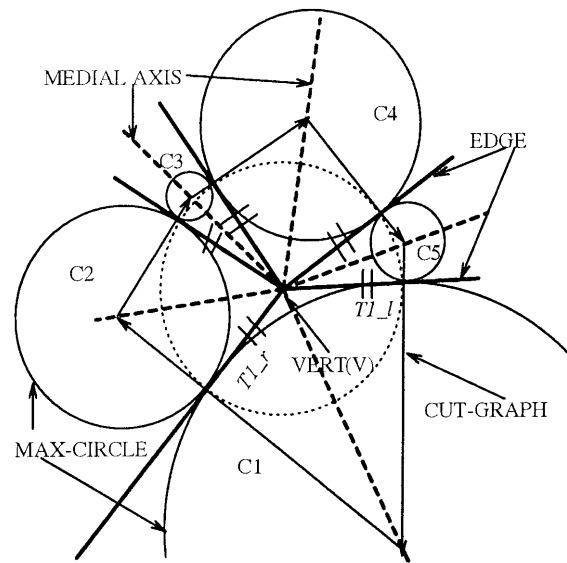


Fig. 8. Laying strips in a 2D patch.

It must be mentioned here that the cut-points are inserted in each MAP-SEG segment, and therefore each division introduced during the punching process inherits the connectivity information associated with the segment MAP-SEG. Connectivity information between the strips is therefore readily established.

The possible shapes of a strip inside a patch is shown in Fig. 9.

The next step in the algorithm is to generate the nodes and the elements in the all quad mesh. A major portion of the strips can be discretised into quad elements by placing equidistant nodes on the ribs forming a strip. This is illustrated in Fig. 10. As the corresponding ribs in the strips meeting at the medial axis have the same number of nodes ( $N_i$  and  $N_{i+1}$ , respectively, in Fig. 10), the difference between the two is the number of nodes to be placed on the medial axis segment common to the

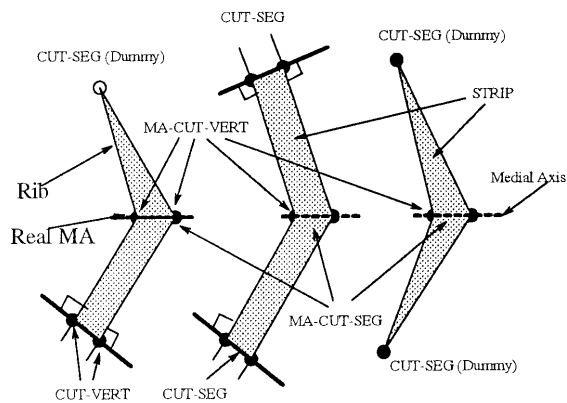


Fig. 9. Possible types of strips.



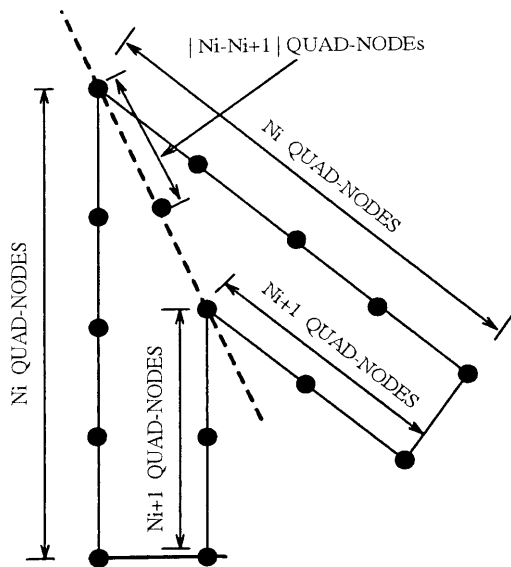


Fig. 10. Placing nodes on ribs to obtain quad element.

strips. This also ensures that there is always an even number of divisions on the ribs forming a strip, thereby enabling an all quad mesh. In fact, except at the junction of a strip and the MA, and at a concave vertex, the placement of nodes on the strip, and their connectivity will always result in quad elements. Additional procedures are required for placing the nodes (and constructing the quad element) at the junction of the strip and the MA, and at concave vertices. The additional procedure involves the merging of two triangles to form a quad element. Again, this is always possible, as there will be two triangles (corresponding to each strip) at the MA and the concave vertex is idealised as a set of even numbers of zero length edges. In some instances (in the vicinity of a branch point of the MAT), some ribs have to be removed to eliminate thin strips (and thereby elements with a bad aspect ratio). Thin strips can occur when the point on an edge corresponding to a branch point on the MA is not the same for two adjacent patches. This is illustrated in Fig. 11.

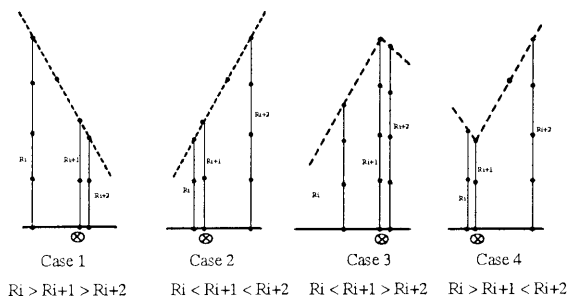


Fig. 11. Removing ribs to eliminate thin strips.

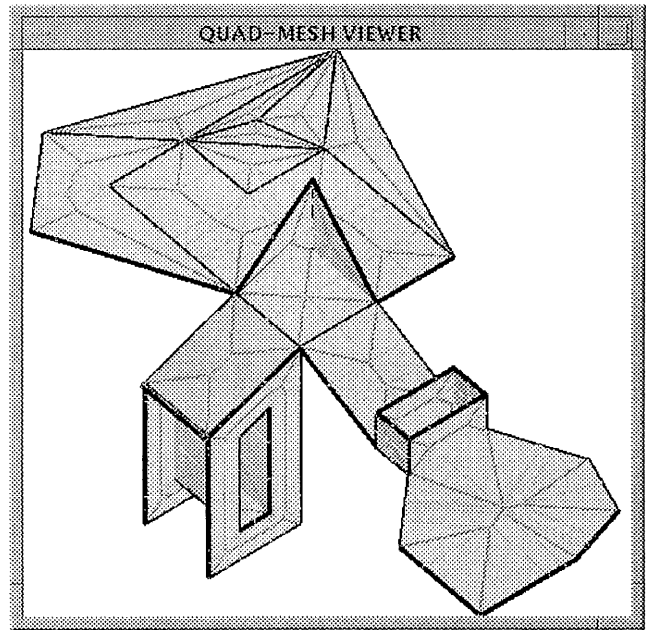


Fig. 12. Set of patches and the two way mapping using Medial Axis in each patch.

## 8. Results and Discussion

The algorithms for generating the midsurface and the all quad mesh have been implemented in a UNIX environment. OpenGL utilities have been used for displaying the results (Figs 12–17). Figure 12 shows a set of patches that form the mid-

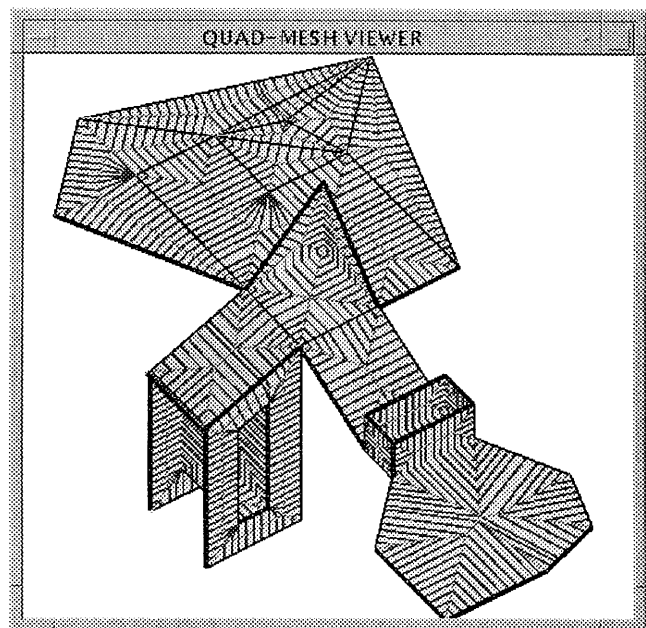


Fig. 13. Set of connected strips on the patches.

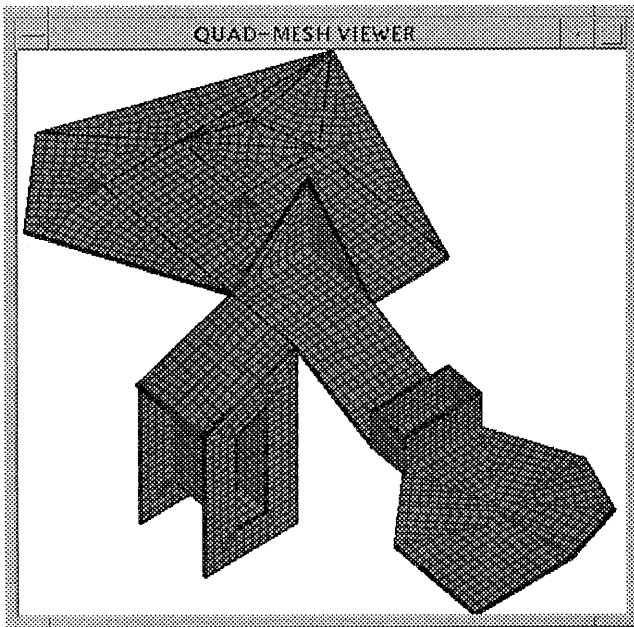


Fig. 14. Quad mesh on the patches.

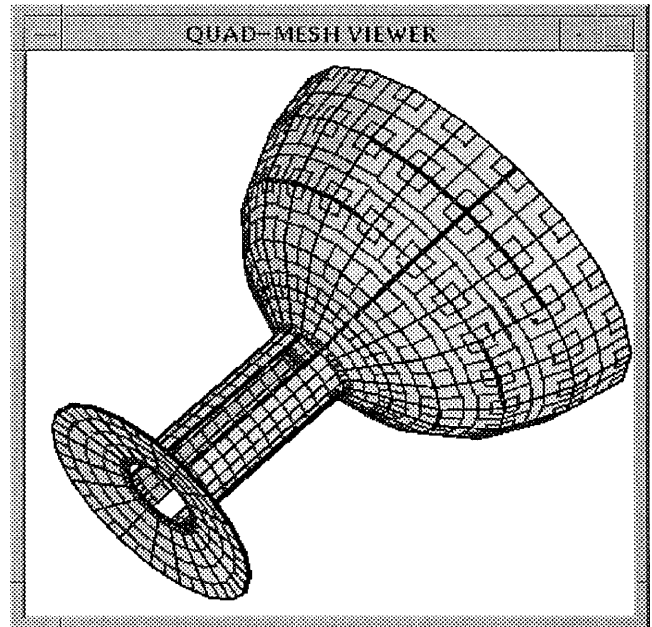


Fig. 16. Set of connected strips.

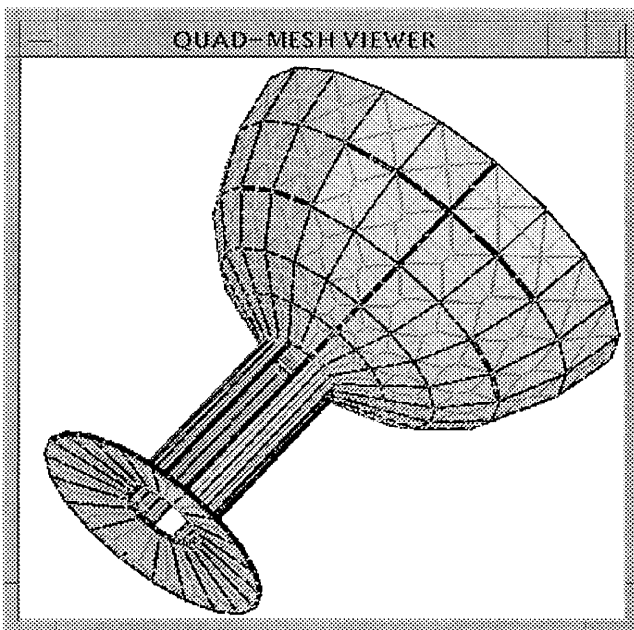


Fig. 15. Two way mapping using Medial Axis.

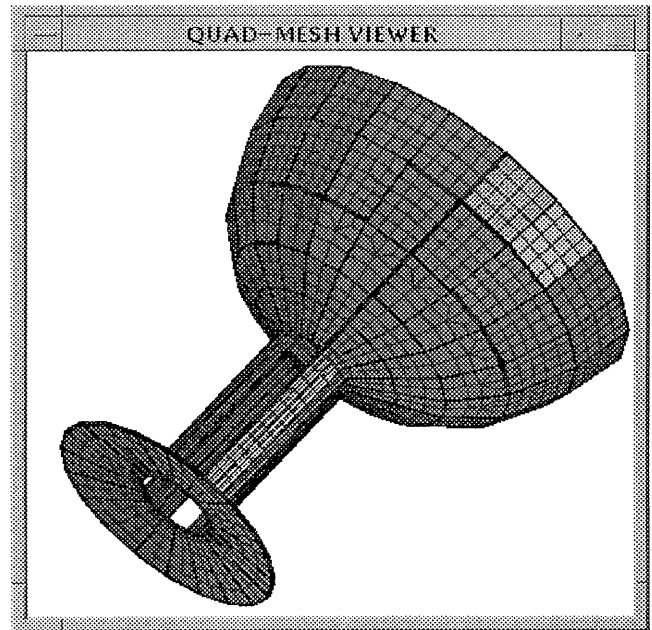


Fig. 17. All quad mesh.

surface of an illustrative example. The composite patch has been selected to highlight the complexities (large variations in size, multiply connected and concave polygons) that occur in the mid-surface abstraction of typical parts. While each patch is planar, the composite surface is three-dimensional. The figure also shows the 2D MAT for each patch in the mid-surface. Figures 13 and

14 show the set of connected strips and the all quad mesh obtained for the part. The final mesh consists of 3190 elements. Figure 15 shows the midsurface of another part. The 2D MAT of each of the patches forming the midsurface is also shown in the figure. Figure 16 shows the strips laid on the midsurface, and Fig. 17 shows the all quad mesh generated on the mid-surface. This mesh contains 1736 elements.

**Table 1.** Average quality of quad elements

Test method	Quad mesh on composite polygon	Quad mesh on cup
Skew angle	5.213°	1.988°
Aspect ratio	1.155	1.364
Taper	0.925	0.951

Tables 1 and 2 present the data on the quality of the mesh obtained by the punching algorithm. The metrics have been obtained from Robinson and Hagenmacher [29], and the thresholds used for calculating the data in Table 2 have been taken from a commercial analysis package. As can be seen, all the quality metrics are very close to the preferred values. It must be mentioned that these numbers are for the meshes as obtained from the algorithm without any post-processing or smoothing.

Once the MA and the two mappings between the boundary and the MA have been established, there are no geometric computations in the punching algorithm. Only a graph like data structure is manipulated to add the nodes and maintain connectivity for the elements. The algorithm is linear in terms of the boundary entities (edges and vertices). The punching algorithm combines the best of the advancing front technique and the decomposition-based approaches. As the nodes are placed first on the boundary and the strips (and subsequently the elements) move from the boundary inward towards the MA, the resulting mesh is sensitive to the object boundary. Also, the strip and the elements terminate at the MA, resulting in a well formed mesh in the interior. Unlike the advancing front technique, no template is required. The use of MA as the termination for each strip avoids the need for interference checks, as required in the decomposition-based approaches.

The algorithm for constructing the midsurface abstraction is dependent on the coordinate system chosen for the discretization. While we believe that using minimal enclosing rectangles may make the resulting skeleton coordinate system invariant,

**Table 2.** Percentage of elements crossing threshold values

Test method	Threshold value	Quad mesh on composite	Quad mesh on cup polygon
Skew angle	$\geq 30^\circ$	1.693%	3.101%
Aspect ratio	$\geq 5$	0.188%	0.183%
Taper ratio	$\leq 0.8$	11.787%	6.819%

currently we have no proof for this hypothesis. As mentioned earlier, the choice of the resolution of the grid used in the thinning algorithm currently requires user intervention at a level that is non-trivial for industrial strength parts.

The punching algorithm can be extended to 3D for all hexahedral mesh generation. The two-way mapping between the object boundary and the MAT in 3D space can be established by projecting a medial surface on the object boundary, resulting in the partition of the faces forming the boundary. The punching algorithm described here would generate a quad mesh on these patches. For all hex mesh generation a new algorithm would lay columns from these patches to the corresponding MAT patches identified by the two-way mapping. Hexahedral elements are then constructed from the boundary to the medial surface in all columns. Merging at the medial axis remains an open question for the completion of robust hexahedral mesh generation.

In the present implementation, all the elements in the quad mesh are roughly the same size. Usually, a finer mesh is desired at sharp corners and concave vertices. Since the radius function associated with a MAT point is a measure of the extent of the object at that point, the radius function can be used to size the mesh adaptively.

Since the MAT is a global property of the part geometry, any change in the object geometry requires the reconstruction of the MAT from scratch. This can be expensive when the MAT is used for planning applications, especially when the environment is dynamic. An incremental construction of the MAT has to be explored so that only the portion corresponding to the modified geometry needs to be reconstructed. This will also be useful in analysis where the MAT would be first used to perform feature recognition. The next step could be the suppression of features that are not critical to the analysis. This step would result in local modification of the geometry. The MAT corresponding to the modified geometry has to be constructed to further proceed with mesh generation. An incremental procedure will be very efficient as the feature suppression for analysis takes place in a sequence of several steps of successive refinement of the geometry, followed by analysis.

Maintaining associativity between the MAT and the object geometry is another requirement that will have to be met if the MAT has to find increasing use in automating analysis and planning. We envisage a scenario (in die design or evaluation, for instance) where the MAT may be changed directly in response to evaluation based on the processability. Alter-

nately, a feature attribute could be changed directly by the application domain. This needs to be reflected in both the MAT and the object geometry as instantaneously as possible to maintain the integrity of the geometry. As the topology of the MAT changes with object geometry, maintaining bidirectional associativity (between the MAT and object) is non-trivial.

## 9. Conclusions

The use of skeletons for enabling design evaluation using numerical simulation has been described in this paper as part of a larger argument in favour of skeleton transformation (MAT in particular) as the representation of choice for automatic reasoning in design and manufacturing. Algorithms for constructing skeletons of 3D objects, robust MAT for 2D objects and all quad mesh on a set of connected surface patches have been presented, along with results of implementation. An algorithm to construct a MAT of 3D objects and the application of these MATs for other applications are identified as scope for further work.

## References

- Blum, H. (1967) A transformation for extracting new descriptors of shape. In: Whalen-Dunn, W. (ed), *Proceedings, Symposium Models for Perception of Speech and Visual Form*, MIT Press, Cambridge, 362–380
- Kim, D., Hwang, I., Park, B. (1995) Representing the Voronoi diagram of a simple polygon using rational quadratic Bézier curves. *Computer Aided Design*, 27(8), 605–614
- Rezayat, M. (1996) Midsurface abstraction from 3D solid models: general theory and applications. *Computer Aided Design*, 28(11), 905–915
- Gursoy, H.N., Patrikalakis, N.M. (1992) An automatic coarse and fine surface mesh generation scheme based on MAT: Part I: Algorithms. *Engineering with Computers*, August, 121–137
- Price, M.A., Armstrong, C.G., Sabin, M.A. (1994) Hexahedral mesh generation by medial surface Subdivision: Part 1. Solids with convex edges. *International Journal for Numerical Methods in Engineering*, 38, 3335–3359
- Quadros, W.R. (2000) Punching – A New Mesh Generation Algorithm using MAT. Technical Report CARL-IISc-2000(3), Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India
- Storti, D., Turkiyyah, G., Ganter, M., Lim, C.T., Stal, D. (1997) Skeleton-based modeling operations on solids. In: Hoffmann, C., Bronsvort, W. (eds), *Proceedings of the Fourth Solid Modeling and Applications Conference*, ACM Press, New York, 141–154
- Gadh, R., Gursoz, E.L., Hall, M.A., Prinz, F.B., Sudhalkar, A. (1991) Feature abstraction in a knowledge-based critique of designs. *ASME Manufacturing Review*, 4, 115–125
- Subrahmanyam, S., Wozny, M. (1995) Overview of automatic feature recognition techniques for computer-aided process planning. *Computers in Industry*, 26(1), 1–21
- Chern, C., Gursoz, E.L., Prinz, F.B., Hall, M.A. (1990) Geometric abstractions using medial axis transformation. *Proceedings of Manufacturing International*, Atlanta, GA, 51–56
- Kao, J.H. (1999) Process planning for additive/subtractive solid freeform fabrication using medial axis transform. PhD Thesis, Stanford University
- Sherbrooke, E.C., Patrikalakis, N.M., Brisson, E. (1996) An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1), 44–61
- Rourke, J.O. (1993) *Computational Geometry in C*. Cambridge University Press
- Lee, D.T. (1982) Medial Axis Transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4), 362–369
- Persson, H. (1978) NC machining of arbitrarily shaped pockets. *Computer Aided Design*, 10(3), 169–174
- Held, M. (1998) Offset Curves of Curvilinear Polygons. *Computer Aided Design*, 30(4), 287–300
- Srinivasan, V., Nackman, L. (1987) Voronoi diagram for multiply-connected polygonal domains I: Algorithm. *IBM Journal of Research and Development*, 31(3), 361–372
- Sudhalkar, A., Gursoz, E.L., Prinz, F.B. (1996) Box skeletons for discrete solids. *Computer Aided Design*, 28(6–7), 507–517
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W. (1993) Mesh optimization. *Computer Graphics (SIGGRAPH 1993 Proceedings)*, 19–26
- Owen, S. (1999) A Survey of Unstructured Mesh Generation Technology, URL: [www.andrew.cmu.edu/user/sowen/survey/index.html](http://www.andrew.cmu.edu/user/sowen/survey/index.html)
- Talbert, J.A., Parkinson, A.R. (1991) Development of an automatic, two dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definitions. *International Journal for Numerical Methods in Engineering*, 29, 1551–1567
- Chae, S., Jeong, J. (1997) Unstructured surface meshing using operators. *Proceedings of 6th International Meshing Roundtable*, Park City, UT, 281–291
- Nowotny, D. (1997) Quadrilateral mesh generation via geometrically optimized domain decomposition. *Proceedings of 6th International Meshing Roundtable*, Park City, UT, 309–320
- Joe, B. (1995) Quadrilateral mesh generation in polygonal regions. *Computer Aided Design*, 27(3), 209–222
- Joe, B. (1996) Delaunay triangular meshes in convex polygons. *SIAM J Sci Stat Comput*, 7, 514–539
- Tam, T.K.H., Armstrong, C.G. (1991) 2D finite element mesh generation by medial axis subdivision. *Advances in Engineering Software*, 13, 313–324
- Blacker, T.D., Stephenson, M. (1991) PAVING: A new approach to automatic quadrilateral mesh gener-

- ation. *International Journal For Numerical Methods in Engineering*, 32, 811–847
28. Cass, R.J., Benzley, S.E., Meyers, R.J., Blacker, T.D. (1996) Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering*, 39, 1475–1489
29. Robinson, J., Haggemacher, G.W. (1992) Element warning diagnostics. *Finite Element News*, June/August