Submitted to ASME Transactions: Journal of Manufacturing Science and Engineering, December, 1999.

Feature Based Volume Decomposition for Automatic Hexahedral Mesh Generation*

[1]Yong Lu [2]Rajit Gadh [3]Timothy J. Tautges

[1]PhD Candidate, I-CARVE Lab, University of Wisconsin-Madison
yong@smartcad.me.wisc.edu, http://icarve.me.wisc.edu

[2]Professor, I-CARVE Lab, University of Wisconsin-Madison
gadh@engr.wisc.edu, http://icarve.me.wisc.edu

[3]Principle Member Technical Staff, Sandia National Laboratories
tjtautg@sandia.gov, http:// endo.sandia.gov/9225/Personnel/tjtautg/

Contact Address: Yong Lu
1513 University Avenue, #146
Mechanical Engineering Department, University of Wisconsin-Madison
University of Wisconsin-Madison,
Madison, WI - 53706, USA.
E-Mail: yong@smartcad.me.wisc.edu,
Phone: (608) 265-3952 Fax: (608) 262-4814

Abstract:

Much progress has been made through these years to achieve automatic hexahedral mesh generation. While general meshing algorithms that can take on general geometry are not there yet; many well-proven automatic meshing algorithms now work on certain classes of geometry. This paper presents a feature based volume decomposition approach for automatic Hexahedral Mesh generation. In this approach, feature recognition techniques are introduced to determine decomposition features from a CAD model. The features are then decomposed and mapped with appropriate automatic meshing algorithms suitable for the correspondent geometry. Thus a formerly unmeshable CAD model may become meshable. The procedure of feature decomposition is recursive: sub-models are further decomposed until either they are matched with appropriate meshing algorithms or no more decomposition features are detected. The feature recognition methods employed are convexity based and use topology and geometry information, which is generally available in BREP solid models. The operations of volume decomposition are also detailed in the paper. In the final section, the capability of the feature decomposer is demonstrated over some complicated manufactured parts.

Keywords: Feature Recognition, FEA, Solid Modeling, Hexahedral Meshing, Volume Decomposition

---

* Extended version of the manuscript: Yong Lu, Rajit Gadh and Timothy J. Tautges, "Feature Decomposition for Hexahedral Meshing", the proceedings of 1999 ASME Design Engineering Technical Conference, Las Vegas, Nevada, September 12-15, 1999. The paper has never been published or submitted for other publications, in whole or in part.

DISCLAIMER

DISCLAIMER

1. Introduction

FEA techniques are playing an increasingly important role in parts prototyping and design verification. The rise of FEA techniques is one of the major factors in the coming productivity gains (Halpern, 1998). Although considerable progress has been made, FEA is still the bottleneck in design practice. Halpern has shown that the analysis phase has a rather lower speed compared to the design phase (Halpern, 1997). In FEA, preparing the model (meshing, mostly) accounts for the significant fraction of time consumed. Thus, it becomes essential to research on automatic meshing algorithms to generate quality meshing models in less time.

1.1 Motivation

Hexahedral meshing is preferable to tetrahedral meshing for FEA model preparation for several reasons (Benzley et al., 1995). There are a few successful automatic tetrahedral mesh generation algorithms, while research on automatic hexahedral mesh generation is still underway. Some algorithms such as Whisker Weaving (Tautges et al., 1996), Plastering (Blacker and Meyers, 1993) and Boundary-fitting (Schneider, 1996) show promise but they are still a long way from being capable of taking on varied models robustly and generating quality meshes on them.

In the real world of meshing, a model that is difficult to mesh automatically is partitioned manually into smaller meshable volumes that are possibly matched with available automatic meshing algorithms. Usually tool bodies need to be generated for the purpose of decomposition. Creating tool bodies with the correct shape and then placing them in the correct location with the correct orientation can be a very difficult and time-consuming procedure. The decomposition procedure can go very further until proper meshing algorithms are found. It may have to be redone if the decomposition results are rejected in the meshing algorithm selection phase. Consequently the manual decomposition and algorithm matching procedure usually takes a great deal of time (it can account for 90% of the model preparation time for meshing (Tautges et al., 1997)). Thus to mimic the procedure, an automatic decomposition tool integrated with a meshing algorithm assignment schema is introduced. It decomposes an unmeshable model into sub-volumes and matches them with appropriate meshing algorithms. The formerly unmeshable object then becomes meshable. In another view, there are scores of proven algorithms that can take on varying geometry, such as, sweeping (Blacker, 1996) (Lai and Benzley, 1996), mapping and sub-mapping (White et al., 1995), and working versions of Whisker Weaving, Plastering, etc. By decomposition, the complexity of geometry and topology of the original model is reduced, thus the model becomes meshable with well established approaches.

On the other hand, even if the model can be meshed as a whole, there is still great value in terms of both meshing time and quality. The complicated meshing algorithms that take on more versatile geometry are computationally more expensive than simpler meshing algorithms. For example, the meshing speed of Whisker Weaving is 100 hexahedral elements per second, while the speed of Mapping is 10,000 hexahedral elements per second (Blacker, 1994). The difference is very significant. Through decomposition, the model is divided into smaller volumes that can be matched with computationally inexpensive algorithms. In the meantime, time used for decomposition is typically much less compared to general meshing procedures (decomposition can be regarded as a very "coarse" meshing operation). Thus, the amount of time consumed overall can be decreased drastically. What's more, since a local meshing approach can be used for local geometry after partition, it is also possible to greatly increase the quality of meshing.

Generally speaking, more suitable (usually computationally inexpensive) meshing algorithms that minimize meshing time but maximize mesh quality can be chosen on sub-models. Therefore, the reduction in meshing time can be even greater and the overall mesh quality can be even better by avoiding using a more expensive meshing algorithm on the entire model.

In addition, an ideal decomposer performs decomposition with global geometry checking; thus the decomposer can serve as a global optimizer that guides meshers. The selection of the decomposition location and the generation of the cutting surface are not only decided by local geometry but also distant geometry. The compromise between local and distant geometry can result in a better partition of meshing, which can avoid generating bad meshing by local-geometry-opt meshers. However, full global checking is very computationally expensive and difficult to implement in practice. The implementation has to be a compromise. It should be simple yet complex enough to avoid the worst cases.

Therefore, we are achieving:

i. Meshability

ii. Less time with better quality

iii. Global Optimization

1.2 Properties of Decomposition Primitives for Meshing

Suppose M represents the model and B_i is the ith volume in the decomposition set W (n decomposed volumes). Then the volumes decomposed for meshing should have the following properties:

$$B_i \subseteq M, \qquad i \in [1,n]$$
$$B_i \cap B_j = \phi \qquad i, j \in [1,n]$$

This implies:

i. No depression volumes (negative volumes) are allowed.

ii. For a complete decomposition set W and $\sum$ is the operation of exclusive addition, then:

$$M = \sum_{i=1}^{n} B_i$$

The properties come from the fact that set operations of general union, intersection and subtraction are very difficult to implement for meshing entities.

1.3 Decomposition Based Meshing

Some decomposition based meshing algorithms are proposed (Tautges et al., 1997) (Liu and Gadh, 1998) (Liu, 1997) (Price et al., 1995) (Price and Armstrong, 1997) (Sheffer et al., 1998) (Hohmeyer et al, 1995).

ICEM AutoHexa is an object-based hexahedral mesher (Hohmeyer et al, 1995), but it has the following drawbacks: the decomposition primitives are limited to a few types (blocks, rectangles, cylinders, discs, and triangular prisms), and the user must decompose the model manually before inputting the decomposed sets into the mesher.

Price, Armstrong, *et al.* suggest the medial surface for decomposition and meshing (Price et al., 1995) (Price and Armstrong, 1997). A medial surface is the path of the center of an inscribed sphere with maximum radius rolling through the model. The medial surface together with maximum radius can be a geometric representation for solid models. Price *et al.* (1995) use the medial surface to guide decomposition and then mesh decomposed primitives by midpoint subdivision. However, generating the medial surface is practically difficult for general geometry. It is also not sufficiently intuitive to result in good partitioning and meshing.

Instead of using the medial surface directly, Sheffer *et al.* (1998) propose the Embedded Voronoi Graph to guide decomposition. The Embedded Voronoi Graph is an approximation of the Voronoi diagram and the medial surface of bodies. It contains the full symbolic information of the Voronoi diagram and the medial surface. The computing of the Embedded Voronoi Graph is robust compared to the computing of medial surfaces. However, as the medial surfaces based decomposition, the algorithm can bring about over-decomposition of bodies and often is not intuitive.

1.4 Feature Recognition

Extensive research on Feature Recognition (FR) techniques has been performed over the past twenty years (Subrahmanyam and Wozny, 1995) (Sonthi et al. 1997). Traditionally, Feature Recognition is the procedure used to extract regions related to design intents or manufacturing functionalities from a solid model. Generally, techniques for Feature Recognition can be classified into two groups: volume-based and boundary-based, while the boundary-based techniques are further classified as rule-based, graph-based and syntactic based.

Since FR techniques are developed mainly as the supporting tools for design and manufacturing, most of them are not suitable to serve as a feature determination approach for meshing due to one or more of the following reasons:

i. Only limited sets of feature primitives can be recognized for specific applications. This is adequate for design and manufacturing applications since in those domains most applications involve only a small set of features that are functioning in those specific applications. However, this is not suitable for the purpose of decomposition for meshing since only a limited portion of the model can be determined and decomposed.

ii. The primitives set is not intuitive for meshing. Some FR techniques use protrusive and depressive feature primitives. As we addressed previously, the set operation of general union, intersection or subtraction for meshing is very difficult to implement. Thus, the decomposed results with mixed concave and convex primitives are not suitable for meshing.

iii. Geometric models targeted are not BREP based. The BREP model is the primary model supported by most meshers. Algorithms that do not work on BREP are not suitable.

iv. Some recognition algorithms require extra information besides the regular information provided by BREP models. Those FR techniques are not good choices since the extra information they need is not universally available in general BREP models.

Gadh and Prinz (1992) suggest an abstraction of CLoop for feature recognition. CLoop was introduced as a closed set of linked edges with the same convexity. Convexity represents one of the major shape properties and can be generally acquired from BREP models. The CLoops, combined with some rules, can be used to extract both protrusive features like ribs, bosses, etc. and depressive features like holes. In this paper, feature recognition techniques based on the extension of CLoop concept are used in the general feature determination phase.

Hence, as shown in Fig. 1, the Feature Decomposer takes a solid model as input and then outputs meshing features. Meshing features are the sub-volumes matched with specific meshing algorithms. Finally, meshing features would be imported into a mesher for meshing.

2. Features for Decomposition

A Feature Decomposer extracts meshing features from a solid model. The procedure consists of three steps: Feature Determination to determine a decomposition feature, Volume Decomposition to extract sub-volumes, and Meshing Algorithm Assignment to assign appropriate meshing algorithms to sub-volumes (shown in Fig. 1).

In fact, there are two kinds of feature recognition procedures happening here. One is the general determination procedure to recognize decomposition features by using CLoop-based FR techniques; the other is the meshing-specific recognition procedure to determine meshing features with automatic meshing algorithms assigned to sub-volumes. We begin with an introduction to CLoop and Meshing Features.

2.1 Meshing Features

The ultimate goal of Feature Decomposer is to get meshable sub-volumes. These sub-volumes, when matched with appropriate meshing algorithms, become meshing features. They can be sweeping features, mapping/sub-mapping features, plastering features, Whisker Weaving features, etc.

The determination of meshing features is to match an appropriate meshing algorithm to the volume. The "appropriate" meshing algorithm means the least computationally inexpensive one that can be used on the part while maintaining reasonable mesh quality. In the case that a part can be meshable by several different meshing algorithms, the most suitable algorithm is selected from the available algorithms based on both quality and expense.

The topological and geometric requirement of many meshing algorithms is not obvious and can be difficult to be abstracted. The matching pattern of a meshing algorithm has the nature of changing along with the development of the meshing algorithm. For example, the original sweeping algorithm can only take on extrusion features with a single source surface and a target surface. Now it is extended to be sweepable among multiple source and target surfaces. Furthermore, sweeping path is no longer constrained to be linear. The stronger the meshing algorithm, the simpler the pattern may be.

Meshing features determination requires extensive knowledge of meshing. Usually, their definition depends on a specific mesher that takes these meshing features as input. The set of meshing features will be expanded when new algorithms are introduced and reduced when some algorithms become obsolete or are merged with one another. Meshing algorithm assignment is the interface to bridge a volume decomposer and a mesher.

2.2 CLoop for Decomposition

CLoop was defined as a closed set of linked edges with the same convexity (Gadh and Prinz, 1992). For the purpose of decomposition, Liu and Gadh extended the definition of CLoop to be an open or closed link of edges by introducing PLoop and HLoop (Liu and Gadh, 1998) (Liu, 1997). This paper relaxes the definition of PLoop and HLoop further and introduces SLoop, a kind of CLoop with mixed convexity.

An edge can be classified as Concave Edge, Convex Edge, Neutral Edge and Hybrid Edge based on edge convexity. Specifically, hybrid Edge is the edge with mixed convexity.

For the purpose of decomposition, we classify CLoop as below:

- *Pure CLoops.* Pure CLoop is a closed link of edges with the same convexity. There are Pure Concave CLoop, denoted as λ^{pcc}, with all edges concave, Pure Convex CLoop, denoted as λ^{pcv}, with all edges convex, and Pure Neutral CLoop, denoted as λ^{pn}, with all edges neutral. For simplicity, we refer to Pure CLoop as PLoop.
- *Pseudo CLoop.* Pseudo CLoop, denoted as λ^{s}, is a closed link of edges with mixed convexity. The edges in the CLoop can be neutral, concave, convex or hybrid. In practice, only a small set of Pseudo CLoops is useful for decomposition. Thus, we set a threshold for the edge angle of the edges that are convex or hybrid. For convex and hybrid edges, only those edges with the edge angle close to 180° are qualified as edges in Pseudo CLoop. In addition, at least one of the edges that form a Pseudo CLoop must be concave or neutral concave. By enforcing these constraints, the set of Pseudo CLoops is much smaller and the decomposition that results in an acute angle can be avoided to some extent. For simplicity, we refer to Pseudo CLoop as SLoop.
- *Hybrid CLoop.* Hybrid CLoop, denoted as λ^{h}, is an open link of edges with the same or mixed convexity. For convex and hybrid edges, similar geometric constraints as with SLoop are enforced to ensure that a limited set of hybrid CLoops, which are suitable for decomposition, is formed. Also, as with SLoop, it is required that at least one concave or neutral concave edge exists in a Hybrid CLoop so that the link of edges is good for decomposition. For simplicity, we refer to Hybrid CLoop as HLoop.

The three kinds of CLoop are illustrated in Fig. 2.

As we discussed previously, depressive (negative) primitives are not in our interests for the purpose of decomposition for meshing. Only concave-like CLoops are required to extract protrusive features, so Pure Convex CLoops and non-concave Pure Neutral CLoops are not considered in our following discussion. For simplicity, when we mention PLoops, from now on, we mean only concave-like Pure CLoops.

2.3 Decomposition with Meshing Features

We discuss two kinds of features so far: meshing features and CLoop features. Meshing features can also be used to guide decomposition. They allow for the extraction of decomposition features that can not be defined by CLoop. Therefore, the model can be decomposed more thoroughly and intuitively. Decomposition with meshing features is more computationally expensive than decomposition with CLoop. Usually, we begin with CLoop for decomposition. For those sub-volumes not matched with meshing algorithms, meshing features are used directly to guide further decomposition. Since the searching space becomes smaller after partitioning with CLoop, decomposition with meshing features can work much more efficiently than taking on the entire body in the very beginning.

3. Feature Decomposition

There are three stages in our approach: CLoop based feature recognition, volume decomposition with imprints propagated properly, and meshing features determination. The model is decomposed recursively until there is no further decomposition necessary with all sub-pieces matched with proper meshing algorithms.

During the feature recognition phase, CLoop is extracted and the decomposition feature is determined. Specifically for HLoop, one more step is needed to close the open link with neutral edges generated by traversing surfaces.

During the volume decomposition phase, a cutting surface is formed based on a bounding CLoop and then the volume is separated into two sub-volumes with the cutting surface. Two new bodies are generated with every cutting. Four steps are performed during the volume decomposition phase: forming cutting surfaces, separating volumes, generating decomposition trees and body relationship graphs, and propagating imprints.

During the meshing feature determination phase, proper meshing algorithms are assigned to sub-volumes according to the set of meshing features available from a mesher. Possible meshing sequences are also suggested.

We describe them in detail in the following sections.

3.1 Determination of CLoop and Generating Cutting Surface

CLoop determination is a graph searching problem. Both PLoop and SLoop are a closed link of edges, while HLoop is an open link of edges. The procedure to search PLoop and SLoop is very much the same except for the different requirement on convexity.

3.1.1 Determination of PLoop and SLoop

A PLoop is a closed link of edges sharing the same convexity. A depth-first searching algorithm is used to find a PLoop. In brief, when searching a pure concave CLoop, the detection begins with a concave edge and it serves as the current tracing edge and is inserted into a list that represents the traversal path. Among all edges connected with it, one of the concave edges is chosen as the next current edge and the tracing procedure is repeated recursively. A PLoop is formed when the group of edges in the traversal path forms a cycle. A marking schema is developed to avoid PLoop duplicating by detecting multiple PLoops.

In essence, the procedure to determine a SLoop is the same as a PLoop. A SLoop can define some features that are impossible to define by a PLoop or a HLoop. For example, it can be used to determine a fillet feature. As shown in Fig. 2, two SLoops can be extracted. One of them bounds a fillet-type shape.

3.1.2 Constructing Cutting Surface for PLoop and SLoop

After retrieving a PLoop or SLoop, a cutting surface, which is used for volume separation, is generated by bounding the CLoop.

The most straightforward method is to fit the link of edges with a single surface. If all of the edges lie on one single analytic surface (the simple analytic surface is a plane), a single analytic cutting surface can be formed. Otherwise, theoretically a single Bspline surface can be generated to cover the chain of edges. However, in practice, the construction of a Bspline surface for a general layout of a CLoop is computationally expensive and can be difficult. Moreover, the resulting cutting surface might be unusual and lead to a possibly bad cutting.

In general, multiple cutting patches rather than a single cutting surface are generated to fit one CLoop. A blending approach was introduced for the purpose (Liu, 1997). In this method, adjacent edges that share planes are grouped together to form planar cutting patches. The left edges that don't generate planar patches are then grouped with the artificial edges, which are created when forming those planar cutting patches, to be fitted with a Bspline patch.

Another approach called the "natural fitting" algorithm is developed in this paper. Fig. 3 illustrates different decomposition results achieved by the two algorithms over a simple part. In this example, a cube sits at a corner of the part. With the blending algorithm, four cutting patches are generated and the cut-off will not be a cube. The acute angles created by the decomposition will bring difficulty to meshing (Fig. 3 I). In contrast, Fig. 3 II shows the decomposition results achieved using the natural fitting algorithm. The shape of the cube is intact. The two separated sub-volumes are kept prismatic and thus can be easily meshed by the sweeping (Blacker, 1996) (Lai and Benzley, 1996) or the sub-mapping (White, 1995) algorithm.

The natural fitting algorithm tries to keep the original geometry intact and manages to avoid Bspline surface when possible. Limited by the length of the paper, the algorithm is not detailed here. Fig. 4 shows the exact sequence of cutting patches generation when decomposing the part in Fig. 3 with the natural fitting algorithm.

Fig. 5 is another example of decomposition with the natural fitting algorithm. Two planar patches and one cylindrical patch are formed. The corner object is cut off smoothly.

3.1.3 HLoop Determination and Cutting Surface Fitting for HLoop

HLoop determination is more complicated than PLoop and SLoop determination because HLoop is an open CLoop and one more step is needed to complete it. There are two steps in HLoop determination: i) getting the open link of edges; ii) traversing the lateral faces between the two ends of the link and generating neutral edges on the lateral face to complete the link.

The step of getting an open link of edges for HLoop is similar to determining PLoop and SLoop. For PLoop and SLoop, a CLoop is formed when the traversal path ends up with a cycle. For

HLoop, it is formed when the traversal path ends at a vertex where no more successive edges with required convexity can be added into that path.

Right now, there are two algorithms to complete HLoop and generate cutting surfaces: the simple extending algorithm (Liu, 1997) and the natural extending algorithm. The simple extending algorithm only uses the planar surface for face traversal and neutral edge generation. In many cases, the cutting surfaces generated by the simple extending algorithm are not good for decomposition and meshing. Fig. 6 shows different results achieved by the two algorithms.

In Fig. 6, the cutting surface created using the simple extending algorithm is a Bspline surface. The cut is not good since the two sub-volumes separated by the cutting surface are not extrusion shapes anymore. In contrast, the natural extending algorithm works very well: three cutting patches--two planar and one cylindrical--are generated. The extrusion nature of the shapes is kept intact. Quality mesh is expected by using fast meshing algorithms such as sweeping or mapping/submapping.

The natural extending algorithm for HLoop is not detailed here. Issues involved are surface intersection, curve trimming, edge grouping and surface creation.

In this algorithm, a set of surfaces called lateral surfaces, which are intersected with the extending surfaces that originate from the HLoop, is traversed. The traversal procedure is recursive. It stops until the original HLoop is completed through the newly created artificial edges. Obviously the lateral surfaces can be more than simple surfaces with single bounding loops. Fig. 7 illustrates different cases of complicated lateral surfaces. If all the lateral surfaces are simple, there is only one traversal thread through the whole traversal path. Otherwise, the traversal path will be split into multiple traversal threads when entering a complicated lateral surface and merged thereafter when leaving the lateral surface. Fig. 8 gives an example of the traversal path splitting and merging.

3.1.4 Cutting Surface Tailoring

At the beginning of generating cutting surfaces from CLoops, only exterior geometric information is taken into account. There can be holes or depressions inside the body. In these cases, the cutting surface needs further refinement with all holes tailored out. Fig.9 gives an example of cutting surface tailoring.

3.2 Volume Decomposition and Imprint Propagation

During decomposition, each cutting step divides one body into two sub-bodies. Thus, a binary tree is the natural representation of a decomposition procedure. It stores decomposed bodies and records the cutting sequence.

As two sub-bodies are separated, the mesh on the cutting surface of the two bodies needs to be compatible. Thus an imprint, which represents the compatible region, is left on each sub-body decomposed. It will enforce the compatibility of the mesh within it between the two sub-bodies. When an imprint is split during further decomposition, a "split" event is generated to inform the other body that shares the touching surface to split the imprint too. A fundamental decomposition tree doesn't hold the exact relationship among sub-bodies. A body relationship graph (DRG) is thus built with "connectivity" edges, representing the "connectivity" of separated sub-volumes, added in the decomposition tree. If an imprint is split, the affected body is traced from DRG and the imprint on that body is then updated.

3.3 Meshing Algorithm Determination

The set of meshing features is heavily dependent on a specific mesher. The determination procedure is computationally expensive, so the pattern for matching should be well abstracted and the assigning algorithm should be well designed. Currently, the effort to research automatic meshing algorithm assignment is ongoing at Sandia National Laboratories.

4. Results

The implementation is based on ACIS (one of the leading 3D modeling kernels) and is ported to CUBIT (the 3D hexahedral meshing toolkit developed by the CUBIT group at Sandia National Laboratories). Fig. 10, Fig. 11, Fig. 12 and Fig. 13 show some of the automatic decomposition results achieved so far. After decomposition, a large portion of these models could be meshed by computationally inexpensive meshing algorithms such as sweeping, mapping and sub-mapping.

Fig.14 gives an example of a rather complicated cutting surface that is generated when decomposing a challenging test case from Sandia National Laboratories.

5. Conclusions and Future Work

This paper presents feature decomposition as part of the effort for the feature based hexahedral-meshing methodology. There are three stages in feature decomposition: feature recognition to determine decomposition features, volume decomposition with imprints propagated properly, and meshing features determination.

The CLoop based feature recognition technique is used to determine the decomposition feature. CLoop was defined as a closed set of linked edges with the same convexity. For the purpose of decomposition, the definition of CLoop was extended to be an open or closed link of edges by introducing PLoop and HLoop. This paper further relaxes the definition of PLoop and HLoop and also introduces SLoop, a new kind of CLoop with mixed convexity, which is helpful to find the fillet-type feature. These extensions allow for the extraction of features that are difficult to determine with the previous CLoop conception.

The natural fitting algorithm to generate a cutting surface for PLoop and the natural extending algorithm to complete HLoop and generate a cutting surface for HLoop are introduced. The two approaches improve decomposition capability and tend to generate sub-volumes that are more suitable for meshing. During the volume decomposition procedure, a body relationship graph (DRG) is constructed to guide imprints to propagate through the decomposed volumes.

Decomposition features become meshing features when bodies decomposed are matched with appropriate meshing algorithms. Meshing features are then imported into a mesher to get the final mesh.

The approaches introduced use only general topology and geometry information, which is available in CAD models generated by most solid modelers and CAD/CAM software. Code is designed and programmed in the manner of self-adaptation and error-tolerance to achieve high automation when taking on varying geometry. Decomposition results are demonstrated over manufactured parts.

There are some issues that are undergoing further investigation:

i. Cutting Surface Tailoring: the surface can be analytical or free form.

ii. Meshing algorithms directly guided decomposition: the decomposition result can be more intuitive for meshing and more cuttings can be achieved.

iii. Cutting sequences optimization.

6. Acknowledgement

The authors gratefully acknowledge the financial support received from Sandia National Laboratories (sponsored by the US Department of Energy under Contract No. DE-AC04-94AL85000), National Science Foundation GOALI (DMI 9708531), National Science Foundation Engineering Research Equipment (DMI 9622665) and National Science Foundation Early Career Development (DMI 9501760).

7. Reference:

Benzley, S. E., Perry, E., Merkley, K., Clark, B. and Greg S., 1995, "A Comparison of All Hexahedral and All Tetrahedral Finite Element Meshes for Elastic and Elasto-plastic Analysis", Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, p. 179-191.

Blacker, T. D. and Meyers, R. J., 1993, "Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm", Engineering with Computers, V. 9, p83-93.

Blacker, T. D. et al., 1994, "CUBIT Mesh Generation Environment, V. 1: User's Manual", SAND94-1100, Sandia National Laboratories, Albuquerque, New Mexico.

Blacker, T. D., 1996, "The Cooper Tool", Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, p.13-29.

Gadh, R. and Prinz, F. B., 1992, "Reduction of Geometric Forms Using the Diferential Depth Filter", Computer-Aided Design, V.24, N.11, P.583-598.

Halpern, M., 1997, "Industrial Requirements and Practices in Finite Element Meshing: A Survey of Trends", Proceedings of 6th International Meshing Roundtable, Sandia National Laboratories, p.399-411.

Halpern, M., 1998, "Enhancing CAE Effectiveness: The Next Steps", Proceedings of 7th International Meshing Roundtable, Sandia National Laboratories, p3.

Hohmeyer, M. E. and Christopher, W., 1995, "Fully-Automatic Object-Based Generation of Hexahedral Meshes", Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, P.129-P.138.

Lai, M., and Benzley, S. E., 1996, "A Multiple Source and Target Sweeping Method for Generating All Hexahedral Finite Element Meshes," Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, p.217-225.

Liu, S.-S. and Gadh, R., 1998, "Finite Element Hexahedral Mesh Generation via Decomposition of Swept and Convex Basic LOgical Bulk (BLOB) Shapes", Journal of Manufacturing Science and Engineering, ASME Transactions, Vol. 120, No. 4, pp.728-735.

Liu, S.-S., Dr. of Philosophy, Mechanical Engineering, 1997, "The Definition And Extraction of Shape Abstractions for Automatic Finite Element Hexahedral Mesh Generation".

Price MA, Armstrong CG and Sabin MA, 1995, "Hexahedral mesh generation by medial surface subdivision: part I. Solids with convex edges", International Journal of numerical methods in engineering, V. 38.

Price M.A., and Armstrong, C.G., 1997, "hexahedral Mesh Generation by Medial Axis Subdivision: Part II. Solids with Flat and Concave Edges", International Journal of numerical methods in engineering, V. 40.

Schneider, R., 1996, "A grid-based algorithm for the generation of hexahedral element meshes", Engineering with computer, V.12, No. 3-4, p. 168-177.

Sheffer, A., Etzion M., Rappoport A., Bercovier M., 1998, "Hexahedral Mesh Generation using the Embedded Voronoi Graph", Proceedings of the 7th International Meshing Roundtable, Sandia National Laboratories, p.347-p.364.

Sonthi, R., Kunjur, G., and Gadh, R., 1997, "Shape Feature Determination using the Curvature Region Representation", Proceedings of Fourth Symposium on Solid Modeling and Applications, Sponsored by ACM SIGGRAPH, Atlanta, Georgia, pp. 285-296.

Subrahmanyam S. and Wozny M., 1995, "An overview of automatic feature recognition techniques for computer-aided process planning", Computers in Industry 26, I-21.

Tautges, T. J., Blacker, T. D., Mitchell, S., 1996, "The Whisker Weaving Algorithm: a Connectivity-based Method for Constructing All-hexahedral Finite Element Meshes", International Journal of Numerical Methods in Engineering, V. 39, p3327-3349.

Tautges, T. J., Liu, S.-S., Lu, Y. and Gadh, R., 1997, "Feature Recognition Applications in Mesh Generation", the special symposium on unstructured mesh generation at the 1997 Joint Summer Meeting of ASME/ASCE/SES (McNU '97), Chicago.

White, D. W., Lai, M. and Benzley, S. E., 1995, "Automated Hexahedral Mesh Generation by Virtual Decomposition", Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, p.165-176.

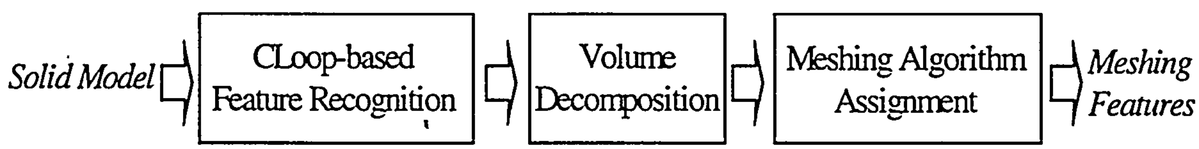


Figure 1, Yong Lu

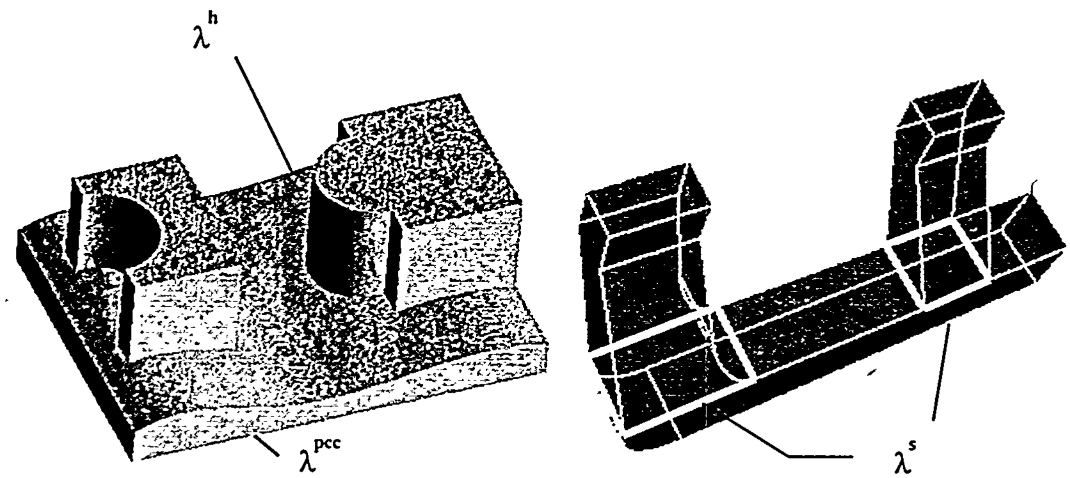


Figure 2, Yong Lu

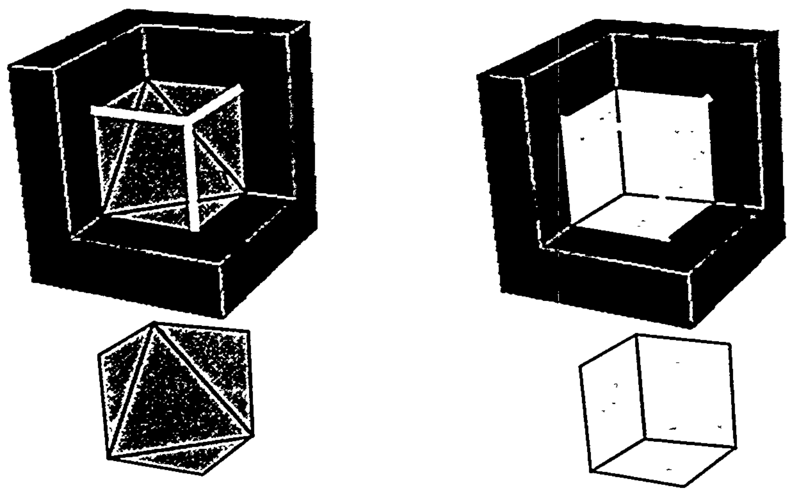


Figure 3, Yong Lu

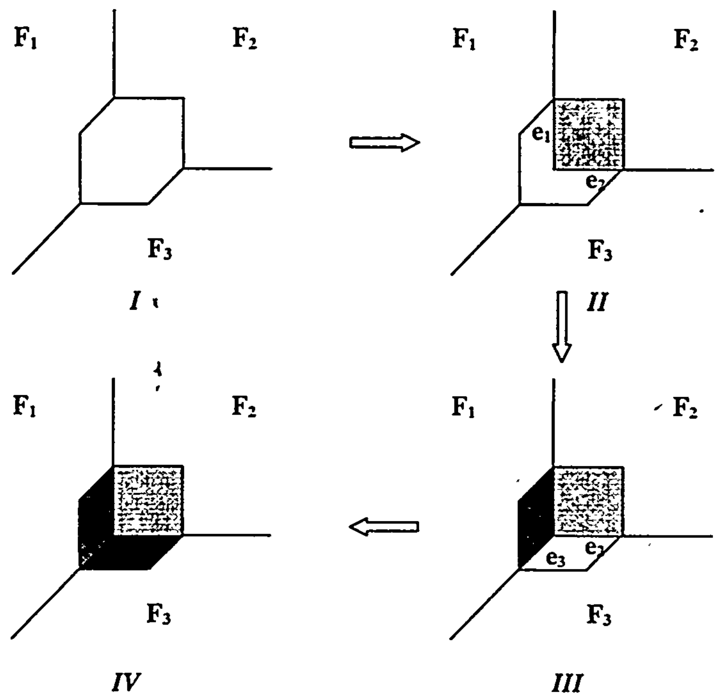


Figure 4, Yong Lu

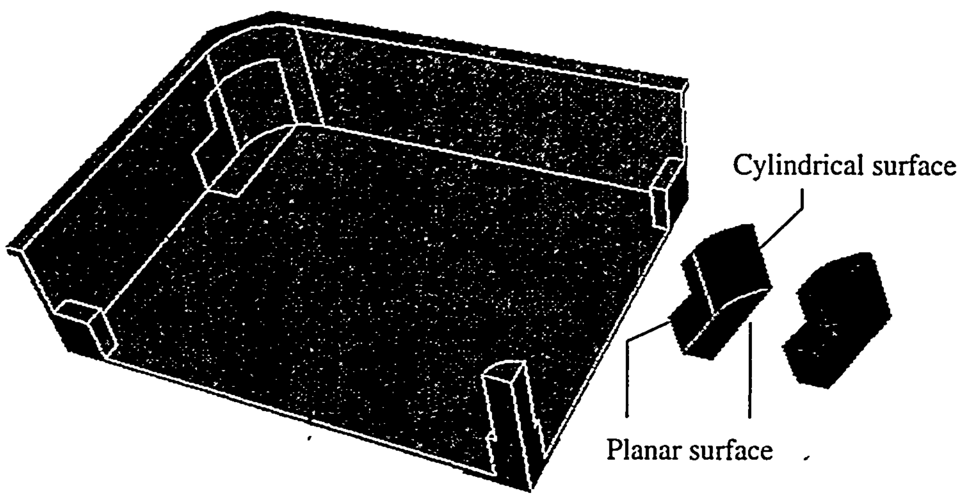


Figure 5, Yong Lu

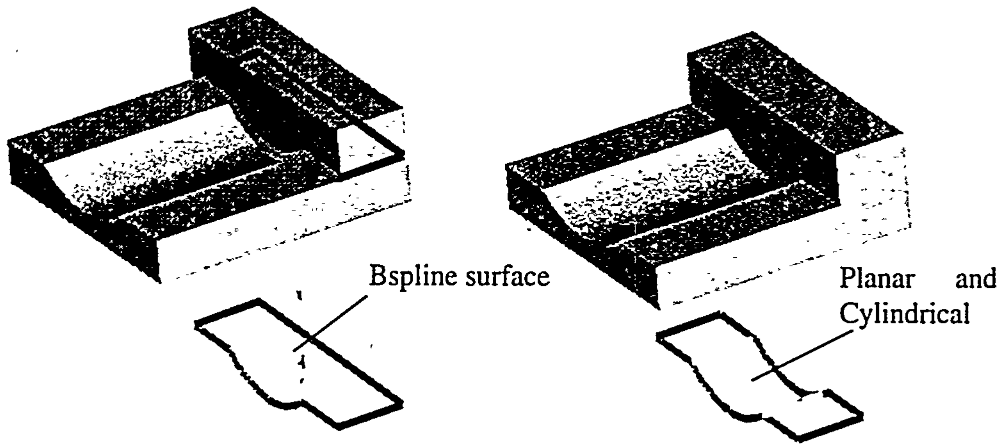
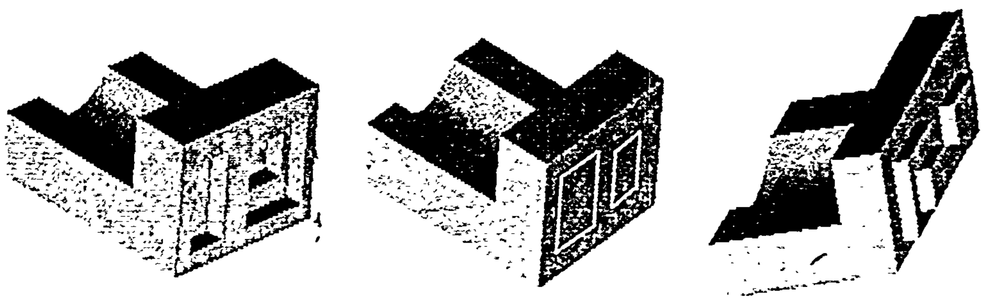


Figure 6 I, Yong Lu

Figure 6 II, Yong Lu



I. concave entities II. imprints III. convex entities

Figure 7, Yong Lu

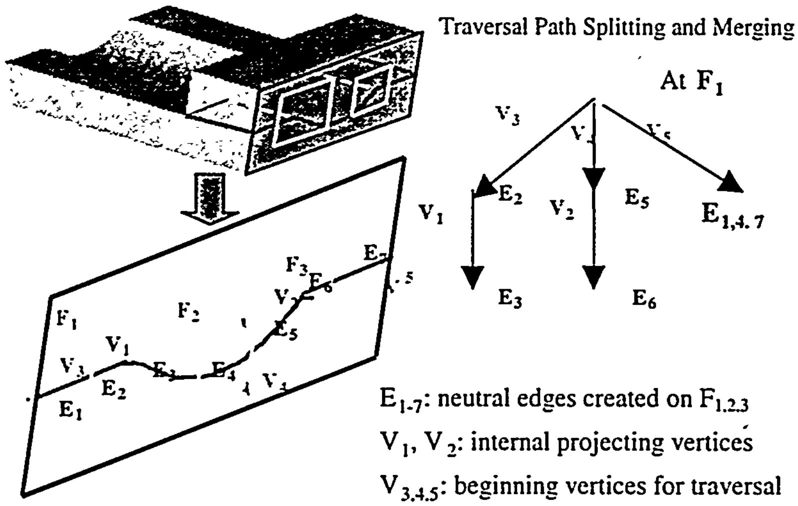


Figure 8, Yong Lu

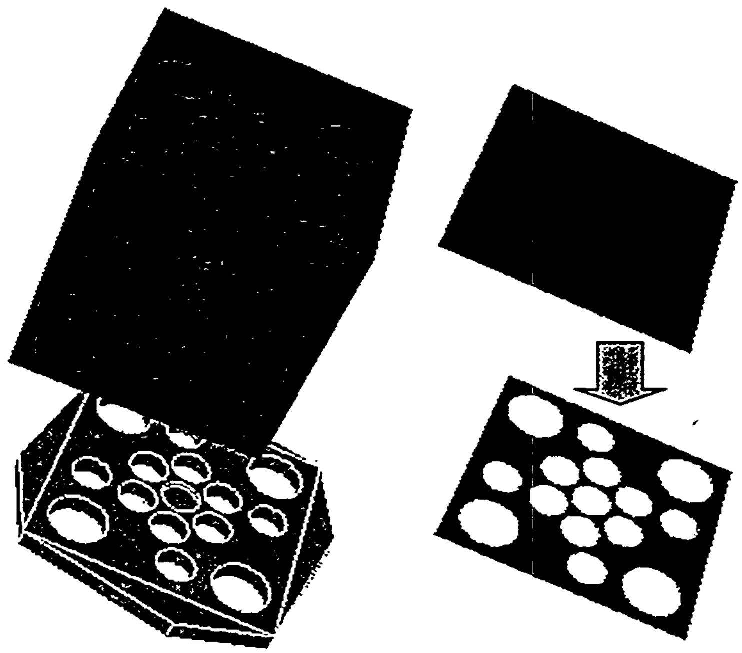


Figure 9, Yong Lu

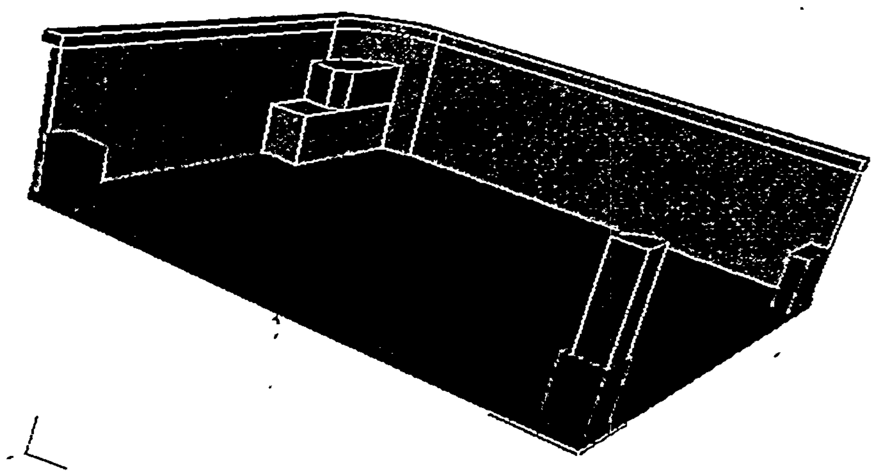


Figure 10, Yong Lu

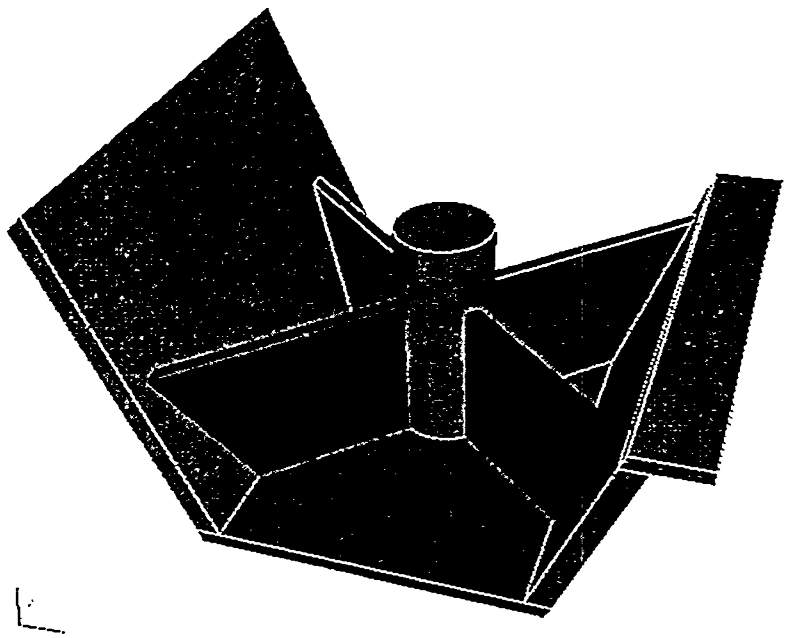


Figure 11, Yong Lu

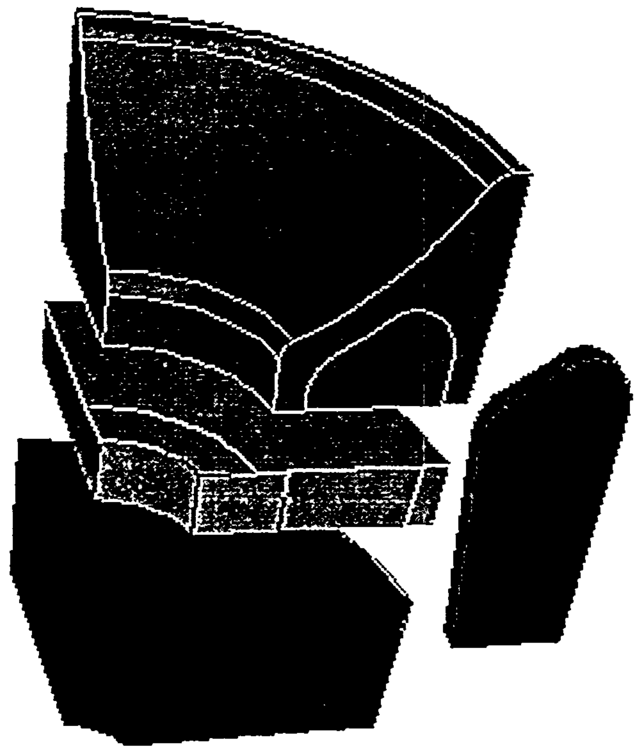


Figure 12, Yong Lu

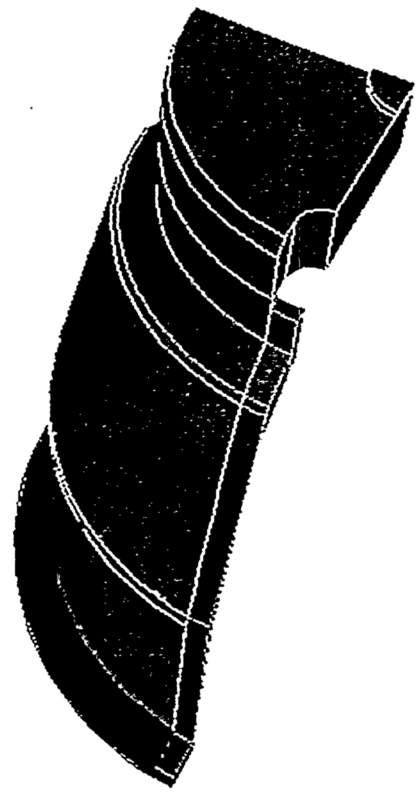


Figure 13, Yong Lu

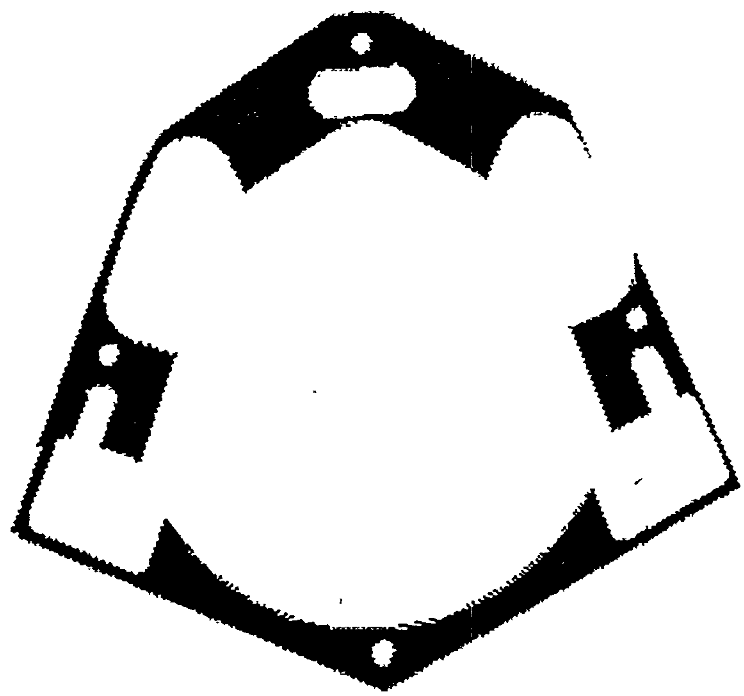


Figure 14, Yong Lu

Figure 1	Feature Decomposer generates Meshing Features
Figure 2	PLoop, HLoop and SLoop
Figure 3	result comparison between the blending and natural extending algorithm
Figure 4	a sequence of cutting patches forming by the natural extending algorithm
Figure 5	an example of natural fitting algorithm for Ploop
Figure 6 I	the simple extending algorithm
Figure 6 II	the natural extending algorithm
Figure 7	cases of complicated lateral surfaces
Figure 8	an example of traversal path splitting and merging
Figure 9	cutting surface tailoring, the part is from Sandia National Laboratories
Figure 10	a test part, from Sandia National Laboratories
Figure 11	a test part, from ALCOA
Figure 12	a test part, from Sandia National Laboratories
Figure 13	a test part, from Sandia National Laboratories
Figure 14	a cutting surface generated for a test part, from Sandia National Laboratories