Multi-Resolution Mesh for Sculptured Surface Machining*

Jiang ZHU** Tomohisa TANAKA** and Yoshio SAITO**
**Department of Mechanical and Control Engineering, Tokyo Institute of Technology, 11-33, 2-12-1, O-okayama, Mekuro-ku, Tokyo, 152-8552, Japan
E-mail: zhujiang@mep.titech.ac.jp

Abstract
Multi-resolution display of highly detailed mesh models has been gaining considerable interest in recent years. However, most researches today are only focus on the geometric similarity. The mesh quality and machining acceptability are less concerned. In this paper, a multi-resolution mesh generation algorithm for sculptured surface machining is presented. The lower-resolution mesh can be used for rough-cut, and the higher-resolution mesh for finish-cut in machining. The proposed algorithm uses iterative edge collapse to simplify the input mesh. The approximation error is controlled by a local error metric VIV (Vertex Importance Value), and a global error bound. This multi-resolution mesh is capable of generating faithful approximations of the original model with high-quality mesh, and also capable of generating rough-cut models with simple geometric complexity and incremental volume, which is suitable for material removal process.

Key words: Multi-resolution Mesh, Mesh Simplification, Sculptured Surface Machining, Rough-Cut Process

1. Introduction

Currently, there are two conventional methods to fabricate 3D models and sculptured surfaces: RP (Rapid Prototyping) and CNC machining. Both of them require detailed geometric models, which are usually created by CAD software or 3D scanning systems. However, with the rapid development of 3D scanning technology, an accurate representation of a 3D model can easily contain a million triangles, which is difficult to render, store, and transmit. In the field of sculptured surface machining, over sampled mesh data will cause much inconvenience and low efficiency for later processing such as slicing or contouring. One approach to narrow this discrepancy is mesh simplification. Mesh simplification reduces the number of triangles needed to represent a model in retaining a good approximation to the original shape and appearance. Multi-resolution mesh provides a series of approximations of the original model with different number of triangles, which can be used in different applications.

Meanwhile, in a material removal process, such as CNC machining, or hotwire cutting\(^{(1-3)}\), the sculptured surface is machined by executing both rough-cut and finish-cut. Since machining time is a function of the tool path length and the shape complexity of machining segment, complex surface will require more machining time compared to simple surface with the same surface area. However, the goal of rough-cut is to remove redundant material in the most efficient manner. There is no need for the tool path to have all the geometric details of the object boundary. Machining efficiency could be improved by using models with different resolutions in different cutting stages, such as, using lower-resolution mesh with simpler geometric complexity when the cutting surface is far away from the...
object boundary, while using higher-resolution mesh with detailed geometric feature when
the cutting surface is close to the object boundary.

In this paper, a multi-resolution mesh generation algorithm for sculptured surface
machining is presented, which is based on edge collapse transformation. The approximation
error is controlled both by a local metric, called VIV (Vertex Importance Value), and by a
global error bound, boundary envelopes. The proposed algorithm is capable of generating
faithful approximation of the original model with high-quality mesh, as well as capable of
generating rough-cut models with simple geometric complexity and incremental volume.

In the past two decades, many researchers have concentrated on the study of
multi-resolution mesh. Several algorithms have been formulated to simplify the mesh and
generate multi-resolution mesh, such as vertex clustering by Rossignac(4), vertex decimation
by Schroeder(5), edge collapse and progressive mesh by Hoppe(6,7), wavelets based analysis
by Lounsbery(8), vertex pair contraction by Garland(9), and simplification envelope by
Cohen(10). It is noticed that almost all the present algorithms are based on the application of
computer graphics, what those algorithms interest are time efficiency and geometry likeness
rather than the mesh quality. Actually, in sculptured surface machining, mesh quality is
important in the same way. However, few algorithms were developed based on the
characteristic of manufacturing.

Recently, the problem of surface mesh improvement has been becoming more and more
important and received a lot of attentions in the field of manufacturing. The present
methods to improve the mesh can be classified into two types: One method is modification
of mesh topology by inserting or deleting mesh nodes, such as Hoppe(7), and Jiang Zhu(1,3),
and the other one is node movement method commonly called mesh smoothing, such as
Savchenko and Hagiwara(11). However, most of these algorithms are not embedded within
the multi-resolution mesh generation algorithm, and they have to be carried out additionally
to improve the mesh quality. In the algorithm presented by this paper, the special
requirements for sculptured surface machining are considered in each transformation
operation. The resulting approximation keeps the original model feature, as well as provides
high-quality mesh, which makes it suitable for machining.

2. Requirements for Input Mesh in Sculptured Surface Machining

In general, the sculptured surface can be fabricated either by a deposition process or by
a material removal process. In deposition process, such as RP, the input mesh is required to
possess a valid topological structure. In such process, the sculptured surface is machined
layer-by-layer; slicing or contouring techniques are widely used to generate the tool path on
each layer. Any topological problems, such as self-intersection, degenerate facets, undesired
holes or flipped normal vectors, would lead to invalid cross-sections, which cannot be
manufactured as layers.

In material removal process, such as surface carving and end milling, the sculptured
surface is machined face by face in the rough-cut process. It requires the triangles of the
input mesh have regular size and regular shape. Since the cutting tools are selected
according to the size of each face(12), machining the surface, whose triangles are almost in
the same size, will require fewer cutting tools and less tool-change time. In order to prevent
the miss-cut of adjacent faces, some places at the corner of a concave face cannot be
machined. The size of the residual part depends on the size of the machining tool and the
inner angle of that corner. Therefore, machining an equilateral triangle is more accurate
than machining a sliver triangle, whose vertices are almost collinear.

Since the purpose of this approach is the application of sculptured surface machining,
mesh quality has the same importance as geometry likeness. The algorithm is developed
based on the above requirements, which provides a practical combination of quality and
machining applicability. In the next section the algorithm proposed by this paper will be
described and the solutions to meet these requirements will also be addressed respectively.

3. Algorithm of Generating Multi-resolution Mesh

A triangle mesh consists of a set of vertices and triangles. Each vertex $V$ specifies the $(x, y, z)$ coordinate of a point in space, and each face $T$ defines a triangle by connecting together an ordered subset of the vertices. In this algorithm only the triangle mesh is considered, and arbitrary mesh can be converted into triangle mesh through a triangulation process. The input mesh is also assumed to be a closed surface without internal void and self-intersection.

The multi-resolution mesh algorithm is based on edge collapse transformation, which was originally proposed by Hoppe (7). A local error metric VIV (Vertex Importance Value), which was developed in the previous work (1), is used to guide the sequence of simplification and control the quality of the approximations. In this algorithm boundary envelope, which was proposed by Cohen (10), is also introduced into the algorithm. By introducing boundary envelope, the algorithm guarantees the global error of the approximations within a specific threshold, and also provides an efficient solution to generate rough-cut models.

3.1. Edge Collapse and Multi-resolution Mesh

![Fig. 1  Edge Collapse Transformation.](image)

As shown in Fig. 1, an edge collapse transformation $\text{ecol}(\{V_i, V_j\})$ unifies two adjacent vertices $V_i$ and $V_j$ into a single vertex $V_j$. Vertex $V_i$ and two adjacent faces $\{V_s, V_j, V_i\}$ and $\{V_s, V_j, V_r\}$ vanish in the process. A position $V_j$ is specified for the new unified vertex. Therefore, the initial mesh $M_n$ can be simplified into a coarser approximation $M_0$ by applying a sequence of successive edge collapse transformations (7):

$$M_{n+1} \rightarrow M_{n} \rightarrow \cdots \rightarrow M_1 \rightarrow M_0$$

$(M_0, M_1, \cdots, M_n)$ is called the multi-resolution representation of mesh $M$.

3.2. Edge Selection According to VIV (Vertex Importance Value)

After the basic transformation is settled, one importance issue remains: how to select the candidate edge to collapse. In Hoppe’s Progressive Mesh (7), the simplification process is carried out by minimizing an energy function, which denotes the complexity and fidelity of the model. It takes much computation cost to generate the multi-resolution mesh. However, in this developed algorithm, the simplification process is guided by an error metric, which is called vertex importance value, or VIV. This error metric, VIV, reflects the local geometric feature of the model, and also helps to eliminate the generation of sliver triangles. Furthermore, it is mathematically simple to compute, and suitable to produce rough-cut models. The method to calculate VIV of each vertex will be described in the next section.

According to the geometric feature of the model, every vertex is assigned with a VIV value. In the algorithm, the vertex with the minimal VIV in the mesh is collapsed first, and it is collapsed with its closest adjacent vertex. Therefore, there are two loops to search the candidate edge to collapse. The outer loop is searching all the vertices in the mesh to locate
the one with the minimal VIV. The inner loop is searching all the edges connecting that vertex to find out the shortest edge. For example in Fig.1, if \( V_i \) has the minimal VIV of all the vertices, after searching all its adjacent vertices, the edge which connects \( V_i \) and its closest neighbor \( V_j \) will be selected as the candidate edge.

The remaining problem is how to place the new vertex after the collapse transformation. One solution is using the optimal placement strategy, and another one is using the original vertices subset as the new vertex position. Considering that complex mesh models usually have tremendous data quantity, using the original vertices subset will highly increase the simplification efficiency. Therefore, the vertex of the candidate edge, which has a higher VIV, is selected as the position of the new vertex.

3.3. VIV Calculation

VIV is used to guide the simplification sequence in this algorithm. Therefore, it should reflect the local model features to control the error of the simplified results, and it should also eliminate the generation of sliver triangles to guarantee the mesh quality of the approximations.

In order to generate a faithful approximation to the original model, the vertex with lower local curvature value can be collapsed first, because it will cause less change to the model feature. Meanwhile, the edge with shorter length can also be collapsed at first time, because it will cause less visual effect to the model. Furthermore, considering the characteristic of sculptured surface machining, collapsing the edge with shorter length can also eliminate sliver triangles. Simply selecting the edge by vertex local curvature will result the simplified approximations keep all the local detailed features, which are unnecessary. However, by selecting the edge only by length, the approximations will be composed of same-sized equilateral triangles while lose most of the local detailed features. In order to obtain a practical combination of quality and machining acceptability, the VIV is given as the product of its local curvature and the length of its shortest connecting edge:

\[
VIV(V_i) = CUR(V_i) \times EL(V_i)
\]

where \( CUR(V_i) \) is the local curvature value of vertex \( V_i \), and \( EL(V_i) \) is the length between vertex \( V_i \) and its closest adjacent vertex.

\[
\rightarrow \cdot \rightarrow = \cos(\angle(T_a, T_b)) = n_a \cdot n_b \quad (0 \leq \angle(T_a, T_b) \leq \pi)
\]

If the dihedral angles of the triangle pairs around one vertex are all close to \( \pi \), the surrounding area of that vertex can be considered as an approximate plane. Obviously, it has a lower local curvature value. Oppositely, if the dihedral angles of triangle pairs around a vertex are sharp, that area must have a relatively high local curvature. Therefore, if vertex
$V_i$ is surrounded by a triangle set $T_a, T_b, T_c, ..., T_k \in T(V_i)$, as shown in Fig.2, the local curvature value of $V_i$ can be approximately represented by the triangle pair, which has the minimal dihedral angles among all its surrounding triangles. In order to unitize the curvature value, it can be described as follows:

$$\text{CUR}(V_i) = \text{Max}\left(\arccos\left(\frac{n_m \cdot n_n}{|n_m||n_n|}\right)\right) \quad (m,n = a,b,..,k, m \neq n) \quad (3-3)$$

### 3.4. Boundary Envelope Computation

Although VIV can generate approximations with high quality, it only provides a local error metric to control the simplification error. In order to obtain a global error bound, which guarantees that the distance between the original model and the approximations is within an absolute threshold, boundary envelope is introduced into this algorithm. It generates two offset-surfaces for an input mesh, one surface on the outside of the mesh, called as outer envelope, and the other on the inside, called as inner envelope. These two offset-surfaces are defined as boundary envelopes, and used to guarantee that the simplification results would never deviate over these boundaries. In this section a brief introduction on computing the boundary envelopes is reviewed, and the difference between the algorithm developed here and Cohen’s approach\(^{(10)}\) is indicated.

In order to create the boundary envelopes of the model, the algorithm offsets each vertex instead of offsetting each triangular facet. For a user specified offset value $\varepsilon$, as shown in Fig.3 (a), the position of each vertex is extended along its normal vector by $\varepsilon$ to create an outer offset-surface; similarly the vertex position is displaced along the opposite direction to create an inner offset-surface. The normal vector of vertex $V_i$ can be calculated by the area-weighted average of the normal vectors of its surrounding triangles, shown by equation (3-4):

$$\vec{N}_i = \frac{\sum n_j A_k}{\sum A_k} \quad (3-4)$$

where $\vec{N}_i$ is the normal vector of vertex $V_i$, $n_j$ is the normal vector of its surrounding triangle, and $A_k$ is the area of each triangle.

In order to preserve the topology of the input mesh, the boundary envelopes are not allowed to have self-intersection, which is illustrated in Fig.3 (b). To meet this criterion, the scale of offset has to be reduced at certain places. Each triangle together with its inner offset and outer offset forms a prism. Using these prisms, the self-intersection of the outer and inner offset-surfaces is detected. If there exists areas of self-intersection in the
offset-surfaces, the offset-surfaces will be scaled around these regions until the intersection disappears. Thus, the offset value around certain area of the model would be smaller than the specified $\epsilon$. The largest permissible offset scale can be quickly found by dichotomy searching. Fig.4 gives examples of inner and outer envelopes computed by different offset values $\epsilon$. $\epsilon$ is assigned as the percentage of the diagonal of model’s bounding box. From Fig.4 it can be found that in certain areas, such as horns and legs of the cow model, the offset values are not as much as in other areas. The offset values of these areas are reduced in order to prevent the self-intersection of the boundary envelopes.

Once the boundary envelopes are created, they can be used to guide the simplification process. Intersection test is performed to each candidate edge, which is selected by VIV according to the algorithm described in the previous section. If the triangles around that edge do not intersect with the boundary envelopes after the edge collapse transformation, the transformation is acceptable. If there occurs any intersection, the transformation is skipped. The simplification process stops till there is no candidate edge to collapse or the simplified approximation reaches a user specified resolution, which is indicated by the number of triangles left in the approximation. In order to increase the computing efficiency, AABB (axis-aligned bounding box) is employed, which can rapidly eliminate most non-intersection conditions. Thus, by keeping the simplified surface within the envelopes, the algorithm guarantees that the simplified surface never deviates over $\epsilon$ from the original surface and the simplified surface does not self-intersect. Fig.5 gives a comparison of the results computed by VIV approach without and with boundary envelopes. The approximations are simplified into the same resolution level. It can be found that the legs of the cow model is degenerating in the approximation without boundary envelopes, while the approximation with boundary envelopes keeps the detailed features quite well.

In Cohen’s simplification envelopes, it iteratively deletes the removable vertices or triangles to keep the approximation in the envelopes, and then triangulates the resulting holes. However, in this developed algorithm a more general transformation, edge collapse, is used. Vertex removal and triangle removal can be regarded as the special cases of edge
collapse. Furthermore, in this algorithm VIV is used to guide the simplification sequence. It provides better intermediate approximations with lower local deviation.

(a) Approximation without Boundary Envelope  
(1500 triangles)  
(b) Approximation with Boundary Envelope  
(1500 triangles)

Fig. 5  Comparison of Simplification Results without and with Boundary Envelopes.

3.5. Validity Check Process for Appropriate Edge Collapse Transformation

In the simplification process, a given edge collapse transformation may potentially introduce some undesirable inconsistencies to the mesh model, and make the approximations unsuitable for sculptured surface machining. This problem can be solved by applying validity checks to a proposed transformation. If these checks fail, this transformation will be discarded. The validity check process for appropriate edge collapse transformation has two main steps. One is to examine the normal vectors of the triangles around the edge to avoid the regeneration of inadequate mesh. The other one is to estimate the triangular compactness to produce regular mesh.

![Fig. 6  Fold Over Condition.](image)

The most common problem is mesh inversion, as shown in Fig. 6. For example, \(\{V_i, V_j\}\) is the candidate edge to collapse. After the edge collapse transformation, the triangle highlighted by the broken line will fold over on itself. This will produce a crack on the model surface, regarded as flipped face. The method to detect this situation is to examine the normal vectors of the faces around \(V_i\) and \(V_j\) before and after the collapse transformation. If a face’s normal vector changes more than certain significant threshold, it is regarded to fail the validity check. In principle, to prevent fold-over phenomenon, careful selection of the threshold is required. In practice, an empirical value \(\pi/4\) is selected as the threshold to make experiments.

Sliver triangles, which have very small angles, are also undesirable in this application. Here a measure of triangular compactness is introduced

\[
\gamma = \frac{4\sqrt{3}A}{l_1^2 + l_2^2 + l_3^2}
\]  

(3-5)

where \(l_i\) are the lengths of the edges, and \(A\) is the area of the triangle. This will assign a compactness of 1 to an equilateral triangle and 0 to a triangle whose vertices are collinear. Using this formula, a collapse is also regarded to fail the validity check, if it produces triangles whose compactness \(\gamma\) are close to 0.
3.6. Algorithm Summary

The specific details of the algorithm have been described, and the algorithm process can be summarized as the following outlines:

1. Read the original model, which is obtained from CAD software or 3D scanning device.
2. Calculate VIV for each vertex.
3. Make an order list for all the vertices by VIV.
4. Generate boundary envelopes of the original model.
5. Repeat the following steps until it reaches the desired approximation, which is the target to be machined.
   a) Select the top vertex from the order list, and take the edge with its closest adjacent vertex as the candidate edge to collapse.
   b) Calculate the position of the new generated vertex.
   c) Check the intersection of the collapse transformation with boundary envelopes. If failed, skip this collapse transformation and go back to step a.
   d) Check the validity of the collapse transformation. If failed, skip this collapse transformation and go back to step a.
   e) Perform the edge collapse transformation.
   f) Recalculate VIV of the vertices affected by this transformation.
   g) Update the order list.

4. Experiment of Generating Multi-resolution Mesh and Error Analysis for the Approximations

4.1. Experimental Results

The algorithm is implemented on a P4 2.8GHz PC with 512M memory, using JAVA language. Since intersection test is necessary for generating the boundary envelopes, this algorithm only accepts the input models without self-intersection. Another program is developed to preprocess the input model; it will test the self-intersection of model and remove the intersection segments.

Many objects have been tested using the developed program. Fig.7 gives examples of the multi-resolution mesh of cow, bunny and horse model, which are generated using different offset values to compute the boundary envelopes. The algorithm outputs these models until there is no valid edge to collapse. From the results it can be found that in the approximations all the features of the original model are kept very well, although highly simplified. Additionally, by combining VIV method together with boundary envelopes, the topological features of the original models are strictly preserved. In the approximations shown in Fig.7(c) (f) and (i), while most detail of the models has disappeared, the basic characteristic feature of the objects are still intact. The major features such as head, legs, and ears are all apparent, without any degeneration.

Meanwhile, since our algorithm is developed focused on the application of sculptured surface machining, the VIV metric always leads the algorithm to produce a regular mesh. From the results shown in Fig.7, it can be found that the algorithm always tries to generate regular triangles to represent the flat surface in the approximations. Moreover, the triangles in the approximations also have a regular size. These virtues make the approximations easy to be machined.
4.2. Error Analysis of the Approximations with the Original Model

In order to evaluate the quality of approximations generated by this algorithm, some error measurement quantifying the notion of similarity is necessary. A metric $E_{\text{max}}$, which measures the maximal distance between the approximation and the original model, and a metric $E_{\text{avg}}$, which measures the average squared distance between the approximation and the original model, are chosen. The approximation error $E_{\text{max}}(M_1,M_2)$ and $E_{\text{avg}}(M_1,M_2)$ between the original model $M_1$ and the approximation $M_2$ are defined as:

$$E_{\text{max}}(M_1,M_2) = \max \left\{ \max_{f_{i,j}^M(M_1)} d_{i,j}(M_2), \max_{f_{i,j}^M(M_2)} d_{i,j}(M_1) \right\}$$  \hspace{1cm} (4-1)

$$E_{\text{avg}}(M_1,M_2) = \frac{1}{k_1 + k_2} \left( \sum_{f_{i,j}^M(M_1)} d_{i,j}^2(M_2) + \sum_{f_{i,j}^M(M_2)} d_{i,j}^2(M_1) \right)$$  \hspace{1cm} (4-2)

where $V(M_1)$ and $V(M_2)$ are the vertices set on models $M_1$ and $M_2$ respectively. $k_1$ and $k_2$ are the numbers of vertices in $V(M_1)$ and $V(M_2)$. The distance $d_{i,j}(M_2)$ is the minimum distance from vertex $V_i$ to the closest face on model $M_2$. 
can be calculated in the same way. These terms are only used for evaluation purposes; they do not play any roles in the simplification algorithm.

Error analyses are carried out for the simplified approximations, generated using VIV alone and VIV with boundary envelopes. From the analytical results shown in Fig. 8, it can be found out that both methods are able to simplify the approximations with small $E_{\text{max}}$ and $E_{\text{avg}}$ errors. Moreover, both $E_{\text{max}}$ and $E_{\text{avg}}$ errors of the approximations generated by VIV with boundary envelopes are smaller than the results by method VIV alone. In the higher-resolution level, the results using boundary envelopes are much better than the other method, since boundary envelopes provide absolute bounds of the approximations’ deviation. In the lower-resolution level, method using boundary envelopes still provides a better performance on both $E_{\text{max}}$ and $E_{\text{avg}}$ analysis. However, with the model is being simplified, method VIV with boundary envelopes has to enlarge the error tolerance to drastically simplify the approximation. The differences of $E_{\text{max}}$ and $E_{\text{avg}}$ between these two methods are not as high as in the higher-resolution level.

Fig. 8 Error Analysis of Simplified Approximation Using VIV without Boundary Envelopes (VIV) and VIV Together with Boundary Envelopes (VIV&BE)

5. A Further Application for Generating Rough-Cut Models

As mentioned before, in the rough-cut stage of machining process, the main goal is to remove redundant material in the most efficient manner. There is no necessity for the tool path to have all the geometric details of the object boundary. The multi-resolution mesh presented in this paper can also be applied for generating rough-cut models. Higher-resolution mesh can be used close to the object boundary, while lower-resolution mesh can be used far away from the object boundary. Most present approaches solve this problem in a 2D manner, offset the slicing curves then simplify them, because it is much easier to handle 2D curves than to generate 3D rough-cut models. However, this algorithm developed here provides an efficient method to generate rough-cut models in 3D manners.

Multi-resolution rough-cut models can be generated following steps:

1. First, each vertex of the original model is displaced at a specific distance along its normal vector. Using the same technique to generate the boundary envelopes, which is described in section 3.4, a non-self-intersection offset-surface outside the original model can be generated.

2. Then simplification of this offset-surface is implemented. Since no miss-cut of the final work piece is permitted in the rough-cut stage, it requires that the rough-cut model should be always larger than the original model to be machined, or should always include the original model. Therefore, using the original model as the inner boundary envelope will guarantee that the simplified offset-surface will never intersect with the original model. Here only one boundary envelope is employed to check the intersection. Because there is no need to consider the shape similarity
between the rough-cut model and the input model, the other boundary envelope outside the offset-surface is not necessary in this application.

By repeatedly applying the above two steps, a series of such approximations with different resolutions will be generated. It will continuously transform the input model into rough-cut models with simple geometric complexity and incremental volume. Following the reverse sequence of generating rough-cut models, the sculptured surface can be machined step by step from the raw material block. Fig. 9 gives examples of rough-cut models, which are generated by applying different values to offset the original model. They can be used in different rough-cut phases to approach the final work piece.

![Rough-cut Models](image)

6. Conclusions and Future Tasks

In this paper, an algorithm for generating multi-resolution mesh for sculptured surface machining has been presented, which is capable of producing faithful approximations of the original model, and also capable of generating rough-cut models with continuous volume increase. Our algorithm uses edge collapse transformation to simplify models. Error metric VIV works together with boundary envelopes to guide the simplification sequence. VIV is used to control the local error of the approximation and guide the algorithm to generate regular mesh, and boundary envelopes are used to provide an absolute global error bound and preserve the features of original model. This algorithm encourages generating regular mesh and guarantees that the approximations are free from self-intersection, flipped-face and co-linear triangles. The experimental results indicate that the approximations generated by this algorithm are qualified for sculptured surface machining.

The Future work will focus on the following aspects:

1. The proposed algorithm offsets all the vertices along their normal vectors to
generate the offset-surfaces. Actually, vertices in sharp region and smooth region can be treated in different ways to generate better offset-surfaces. This is planned to improve by developing some adaptive offset techniques.

2. The algorithm utilizes AABB to fasten the triangle intersection computation. However, there are still many unnecessary intersection checks between non-adjacent triangles. This can be improved by introducing more sophisticated hierarchical structure, such as octree space-partitioning data structure.

3. The rough-cut models and simplified approximations generated by this algorithm will be used for integrated CNC machining by a multi-axis machining center.

Acknowledgement

The author would like to thank the Stanford University Computer Graphics Laboratory, for offering the horse and bunny models used in the experiments. The author would also appreciate Prof. Garland, for offering the cow model on his homepage.

References


