Kasetsart J. (Nat. Sci.) 42 : 385 - 391 (2008)

Approximation of CAD Models Using Skeletonization

Kunnayut Eiamsa-ard

ABSTRACT

Skeletonization is a method to extract shape information of a CAD model with lower dimension. The associate information requirements of the skeleton are different from application to application. For instance, surface reconstruction requires skeleton with detailed geometry such as distances from the surfaces to the skeleton. On the other hand, object recognition requires only skeleton in order to perform similarity comparison. Even though skeletonization has been studied extensively, most of the previous methods are sensitive to noise, time consuming, or restricted to specific 3D CAD models. A practical method for skeletonization from general 3D CAD models using Centroidal Graph is discussed in this paper. The skeletonization process begins with the decomposition of the 3D CAD part model into sub-volumes or cells. The loci of Centroids are then computed for all cells in order to complete the skeletonization process.

Key words: shape representation, skeleton, medial axis transform

INTRODUCTION

Skeleton, also called Medial Axis Transform, has been used in biology for a long time. Medial Axis Transform represents 3D shapes with a series of curves or points, like the skeleton of a human body. This concept has been widely used in pattern recognition, shape analysis, and mesh generation.

Medial Axis Transform can be used to perform geometric reasoning on complex part model because it has lower dimension compared to the model itself. However, finding Medial Axis Transform (MAT) is very computationally intensive and the result can not be used to perform reasoning since the Medial Axis Transform can be R^2 surfaces embedded in R^3 space. Furthermore, the redundant information may be misleading and is difficult to process. Figure 1 shows an example of Medial Axis Transform of a simple 3D part.

An approach, namely Centroidal Graph, is adopted from the work done by Eiamsa-ard *et al*. (2005) to approximate the topological and geometrical information of the part models. The

Figure 1 Medial Axis Transform by Voxelization.

Department of Mechanical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand.
* E-mail: fengkye@ku.ac.th

approximation of skeleton is easier to compute and can be useful. Instead of searching points with equal distance to the surrounding surfaces to form Medial Axis Transform, the centroids of the part on different locations are obtained to form a Centroidal Graph.

Various methods have been studied to compute the shape representations of 3D parts. These methods can be classified into categories as described in the work done by Iyer *et al.* (2004):

(a) ***Invariant/Descriptor based*** – Invariants or descriptors of the 3D shape such as volume, and surface area are used as signatures;

(b) ***Statistic/Probability based*** – A shape function constructed by random sampling of points is used to describe the part. (Osada *et al.*, 2001) Shape histograms are used to approximate and search for 3D model (Ankerst, *et al.*, 1999);

(c) ***Group technology based*** – Two step Group Technology (GT) method is developed by Iyer and Nagi (1997);

(d) ***Graph based*** – Subgraph isomorphism for matching B-Rep graph are employed. (El-Mehalawi and Miller, 2003) Eigenvalues of a model signature graph (MSG) constructed from the B-Rep graph are used as signatures. (McWherter *et al.*, 2001);

(e) ***Feature recognition based*** – Cells and spatial relationships are computed from part decomposition. (Ramesh *et al.*, 2001);

(f) ***3D Object Recognition based*** – Aspect graphs are used for 3D shape searching. (Cry, and Kimia, 2001) Geometric Hashing is used in the work done by Lamdam and Wolfson. (1988);

MATERIALS AND METHODS

We begin the skeleton approximation process with the decomposition of the 3D part model into sub-volumes or cells by using two types of graphs.

Face-Edge Graph

We normally use Face-Edge Graph to represent the connectivity of faces and edges of a geometry model. Graph represents faces and edges by vertices or points and lines by sequence. Figure 2 shows the geometry model connectivity by Face-Edge Graph.

Cellular Adjacency Graph

Cellular Adjacency Graph represents cells connection. Each cell is represented by point or vertex as shown in Figure 3. The faces that connect 2 cells together or face interactions are defined by lines. (The interactions are shown as edges.)

Cell decomposition technique

The decomposition technique is proposed and discussed in detail in the work done by Eiamsa-ard (2005). Figure 4 shows the examples of cell decomposition.

After the decomposition process has been done, the cells are used to find the skeleton of each cell by using Centroidal Graph.

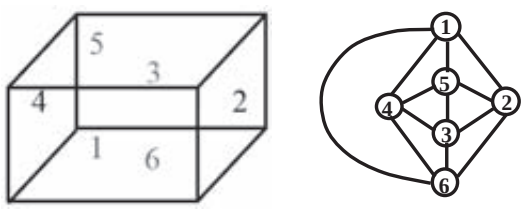

Figure 2 A block and corresponding Face-Edge Graph.

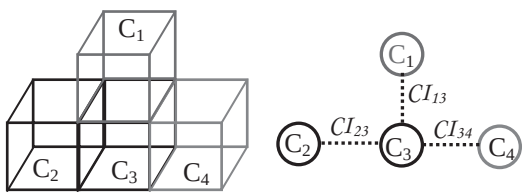

Figure 3 Cellular Adjacency Graph.

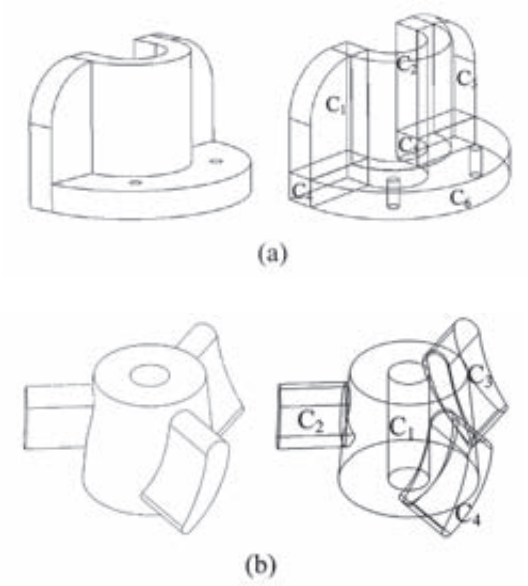

Figure 4 Examples of part decomposition.

CENTROIDAL GRAPH REPRESENTATION

Similar to Medial Axis Transform, Centroidal Graph is also composed of a series of points which are centroids of cross sections of the part at difference locations. A cross section is the intersection of a planar surface with the object. A planar surface can be defined by a position and normal direction.

At particular position, there are infinite directions, Therefore, there are infinite cross sections that yield infinite numbers of Centroidal Graphs for an object. Figure 5 illustrates this situation by showing 3 different Centroidal Graphs along different directions for a simple block.

To avoid this situation, Base- Selection method is proposed in detail in the work done by Eiamsa-ard (2005). However, the Base-Selection Method is described as the follow for the sake of completeness.

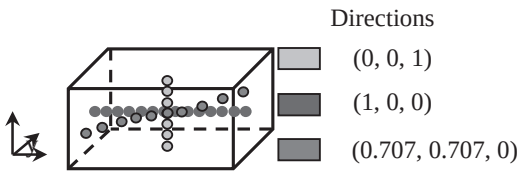

Figure 5 Centroidal Graph along 3 different directions.

BASE SELECTION AND SLICING DIRECTIONS

Base selection is very important as it changes the sequence of how to stack the cells. In this work, the greedy function (F_i) is exploited. Basically, after finishing slicing the current cell, the next cell is determined by Volume Index (VI_i) which needs to be as big as possible.

$$\mathbf{VI_i} = \iiint_{C_i} dv_{Ci} \;/\; \iiint_{P} dv_{P} \qquad (1)$$

The other factors are also considered and included in function $\mathbf{F_i}$: Shape Index ($\mathbf{SI_i}$) and Cut-vertex Index ($\mathbf{CVI_i}$).

$$\mathbf{SI_i} = \iiint_{C_i} dv_{Ci} \;/\; (\Delta x_{Ci} \cdot \Delta y_{Ci} \cdot \Delta z_{Ci}) \qquad (2)$$

where $\Delta\, x_{Ci} = \max(x_{Ci}) - \min(x_{Ci})$

$\Delta\, y_{Ci} = \max(y_{Ci}) - \min(y_{Ci})$

$\Delta\, z_{Ci} = \max(z_{Ci}) - \min(z_{Ci})$

$\mathbf{CVI_i}$ = 1 if C_i = cut-vertex

= 0 otherwise (3)

Therefore, the greedy function $\mathbf{F_i}$ is defined as:

$$\mathbf{F_i} = (w_1 \cdot \mathbf{VI_i}) + (w_2 \cdot \mathbf{SI_i}) + (w_3 \cdot \mathbf{CVI_i}) \qquad (4)$$

w_1, w_2, and w_3 are weights for those factors. The weights can vary propor-tionally. In this work, w_1, w_2, and w_3 are set to be 0.5, 1.0, and 0.25 respectively.

In other words, biggest cut-vertex cells with convex shape are preferred to be sliced first. The cell with the largest $\mathbf{F_i}$ value is selected as the base. The bottom face of the base is preferably picked from the flat convex faces set. (Concavity of the faces is computed from Part P, not cell i^{th}.)

After finishing slicing the current cell, the next cell is selected based on the method described in the earlier work done by Eiamsa-ard (2005). However, the general idea of this method is to select the cell that is adjacent to the previous cell with the appropriate direction to slice. Based on the algorithm discussed in the earlier work, the slicing sequences of sample parts are arranged as shown in Figures 6 and 7.

CENTROIDAL GRAPH COMPUTATION

The skeleton computation of a cell starts at the cross sectional base of cell direction according to cell decomposition technique. Let C_k and L_i be the current cell and the current layer. $\vec{G}_{r(i)}$ is the real centroid of the cross section surface at layer i. Unit $\hat{n}_{r(i)}$ vector is the real normal vector of layer i. The value h is the desired height for each layer.

The 3D cross section algorithm always starts with the prediction step. Point $\vec{C}_{g(i+1)}$ is a guessed point for the next layer i+1 given by:

$$\vec{C}_{g(i+1)} = \vec{G}_{r(i)} + h \cdot \hat{n}_{r(i)} \quad (5)$$

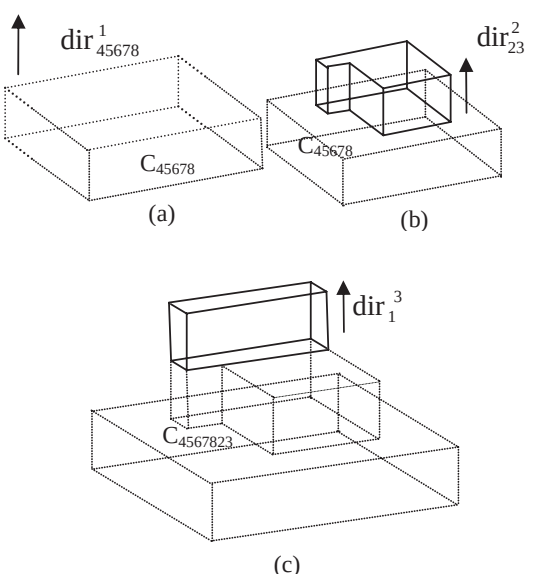

Figure 6 Complete slicing sequence (a)-(b)-(c) of a sample part.

Use $\vec{C}_{g(i+1)}$ as the seed point, and a slice is generated shown with a dashed line in Figure 8. The centroid of this slice ($\vec{G}_{g(i+1)}$) is then calculated. If ($\vec{C}_{g(i+1)} - \vec{G}_{r(i)}$) and ($\vec{G}_{g(i+1)} - \vec{G}_{r(i)}$) are the same, that means the prediction point

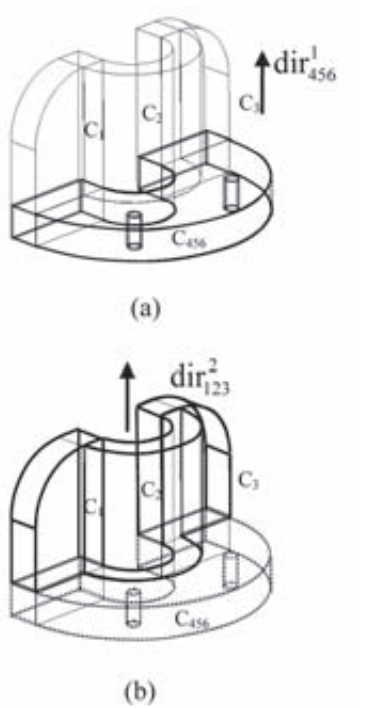

Figure 7 Slicing sequence of another sample part.

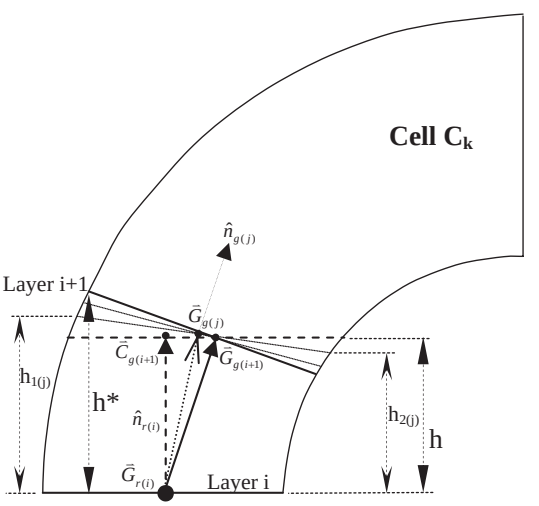

Figure 8 Slicing based on 3D layer.

matches with the real point $\vec{G}_{g(i+1)}$. (In this case, therefore, $\hat{n}_{r(i+1)}$ equals $\hat{n}_{r(i)}$ as well.)

In general, however, $\vec{C}_{g(i+1)}$ and $\vec{G}_{g(i+1)}$ are not the same point. $\vec{G}_{g(i+1)}$ is used as the fixed seed point while the normal vector of the slicing plane is being honed by incrementally shifting the direction from ($\vec{C}_{g(i+1)}$ - $\vec{G}_{r(i)}$) to ($\vec{G}_{g(i+1)}$ - $\vec{G}_{r(i)}$) as far as possible. (Both heights $h_{1(j)}$ and $h_{2(j)}$ have to maintain to be positive.)

Obviously, the value of h* is now bigger than h at the step. Given the h value is always very small, the correction of the height can be simplified as shown in Figure 9.

In order to get the desired height h, the layer is shifted down by y*-y along the vector $\vec{C}_{g(i+1)}$ - $\vec{G}_{r(i)}$. (where y is given in equation 6.)

$$y = \left[\frac{h}{h^*}\right] y^* \tag{6}$$

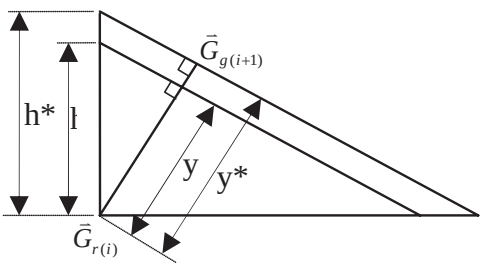

Figure 9 Shifting the layer down to the desired height.

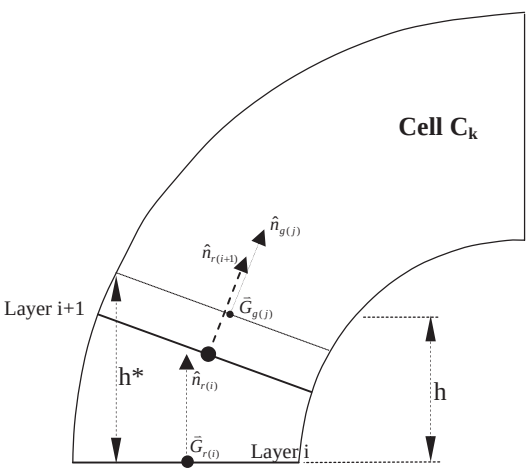

Figure 10 Layer i+1 after adjusting the height.

The normal vector of the final honing slice $\hat{n}_{g(j+1)}$ is used as the predicted normal vector $\hat{n}_{r(j+1)}$ for the next layer i+2. Centroid of layer i+1 denoted as $\vec{G}_{r(i+1)}$ is calculated after the layer is shifted down to be used for the prediction step for layer i+2. (Figure 10)

These procedures need to be repeated until the whole cell has been sliced. Then the series of points can be generated from of every layer. All points are connected to perform a skeleton like. Let S be the Skeleton of a cell.

$$S = \vec{G}_{r(1)} \cup \vec{G}_{r(2)} \cup \vec{G}_{r(3)} \cup \ldots \cup \vec{G}_{r(i)} \tag{3}$$

Centroidal Graph or the loci of centroids is shown in Figure 11.

RESULTS AND DISCUSSION

Skeleton approximation by Centroidal Graph of a Three-D model is the combination of the Centroidal Graphs of all cells in that model. A couple of examples are shown in Figure 12a and 12b.

Our process of skeleton approximation can be separated to 3 steps as following. (a) Cell Decomposition Process, (b) Centroidal Graph Computation Process, and (c) Cell combination Process. These 3 steps are illustrated in Figure 13.

CONCLUSION

This skeleton approximation method is more practical compared to others. Thus, this

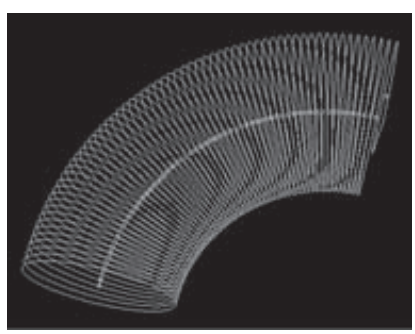

Figure 11 Skeleton of 3D model.

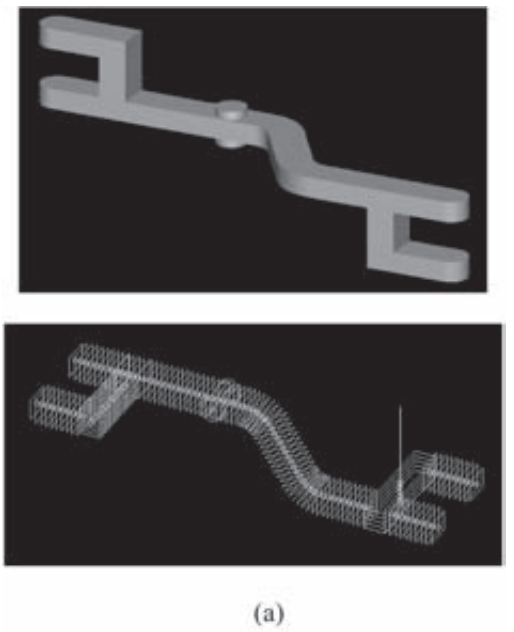

Figure 12 Three-D models and corresponding Centroidal Graphs.

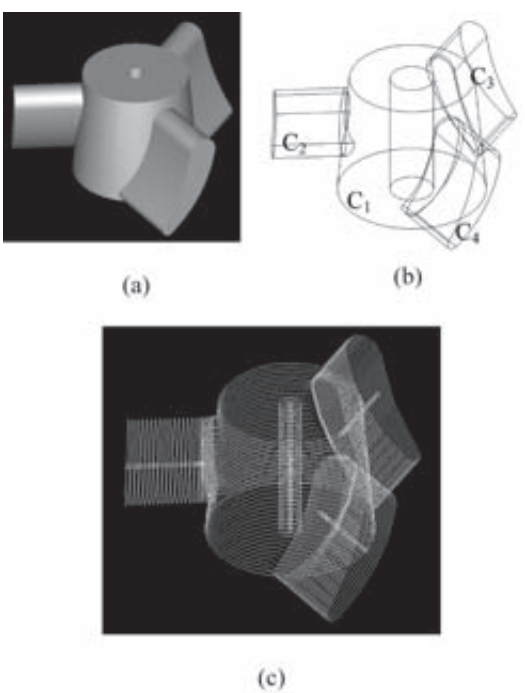

Figure 13 Three steps of Skeleton approximation.

skeleton approximation proposed in the paper is relatively easy to compute. The result can be used in many applications such as to perform object recognition, especially to classify symmetrical and Non-symmetrical model. And also, surface reconstruction is possible with distance information. Loci of centroids can be computed with novel practical method described in this paper.

ACKNOWLEDGEMENTS

The author gratefully acknowledges Laser Aided Manufacturing Process Lab at Missouri University of Science and Technology – Rolla, Missouri, USA for the help in 3D engine.

LITERATURE CITED

Ankerst, M., G. Kastenmuller, H.P. Kriegel and T. Seidl. 1999. **3D shape histograms for similarity search and classification in spatial databases**. Advances in Spatial Databases 6th International Symposium, SSD'99. Hong Kong. China. 1651: 207-228.

Cry, C.M. and B.B. Kimia. 2001. **3D Object recognition Using Shape Similarity-Based Aspect Graph**. in Proceeding of International Conference on Computer Vision, Vancouver, Canada.

Eiamsa-ard, K., F.W. Liou, L. Ren. and H. Choset. 2005. **Building Sequence of Boundary Model in Layered Manufacturing**. Proceeding of IDETC/CIE 2005. California, USA.

El-Mehalawi, M. and R. Miller. 2003. A Database System of Mechanical Components based on Geomatric and Topological Similarity. Part I: Representation. **Computer Aided Design** 35: 83-94.

Iyer, N., S. Jayanti, K. Lou, Y. Kalyanaraman and K. Ramani. 2004. **A Multi-Scale Hierarchical 3D Shape Representation for**

Similar Shape Retrieval. Proceedings of the TMCE 2004. Lausanne. Switzerland.

Iyer, S. and R. Nagi. 1997. Automated Retrieval and Ranking of Similar Parts in Agile Manufacturing. **IIE Transactions on Design and Manufacturing** 29: 859-876.

Lamdam, Y. and H.J. Wolfson. 1988. **Geometric Hashing: A General and Efficient Model Based Recognition Scheme**. Proceeding of International Conference on Computer Vision, Florida. USA.

McWherter, D., M. Peabody, W.C. Regli and A. Shoukofandeh. 2001. An Approach to Indexing Databases of graphs. **Technical Report DU-MCS-01-01**. Department of Mathematical and computer Science. Drexel University. Philadelphia. Pennsyl-vania. USA.

Osada, R., T. Funkhouser, B. Chazelle, and D. Dobkin. 2001. **Matching 3D Models with Shape Distributions**. Proceeding of Shape Modeling International. Genova. Italy.

Ramesh, M., D. Yip-Hoi and D. Dutta. 2001. Feature Based Shape Similarity Measurement for Mechanical Parts. **ASME Journal of computing and Information Science** 1: 245-256.